ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
НАУКИ ИНСТИТУТ АВТОМАТИКИ И ЭЛЕКТРОМЕТРИИ
СИБИРСКОГО ОТДЕЛЕНИЯ РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

Свитов Давид Вячеславович

Оптимизация производительности свёрточных нейронных сетей в системе распознавания лиц

Специальность 1.2.2 —
«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, ведущий научный сотрудник ИАиЭ СО РАН
Нежевенко Евгений Семёнович

Новосибирск — 2023

Оглавление

Стр.

Стр.

Список сокращений и определений

ПК - персональный компьютер.

СНС - свёрточная нейронная сеть.

Средний AP (mAP) - среднее значение метрики Average Precision (средняя точность). Вычисляется как средние по всем классам значение метрики Average Precision, которая вычисляется усреднением значений точности (англ. precision) для различной полноты (англ. recall).

CPU - (Central Processing Unit) центральный вычислительный процессор

GPU - (Graphics Processing Unit) графический вычислительный процессор. Имеет массивно параллельную архитектуру и широко используется для вычисления нейронных сетей.

NPU - (Neural Processing Unit) специализированный подкласс процессоров для эффективного выполнения операций над многомерными тензорами.

FLOPS - (FLoating-point Operations Per Second) количество операций с плавающей запятой, которое вычислитель способен произвести за одну секунду.

API - (Application Programming Interface) протокол взаимодействия между двумя компьютерными программами.

НАП - нейросетевой архитектурный поиск. Семейство подходов для автоматического выбора типа и порядка слоёв в нейронной сети с целью получения модели с наибольшей эффективностью.

Встраиваемые устройства - маломощное специализированное вычислительное устройство на базе микропроцессора, встраиваемое в систему, которой оно управляет. Примером могут служить домофон или система управления камерой наружного наблюдения.

Биометрический вектор - вектор получаемый из изображения лица путём применения свёрточной нейронной сети. При этом считается, что нейронная сеть обучена таким образом, чтобы расстояние между векторами было пропорционально схожести лиц.

Софтмакс с отступом - модификация функции Софтмакс добавлением константы к углу между вектором примера и центром класса, что позволяет увеличить разнесение получаемых векторов в пространстве.

Квантование - обучение нейронной сети с малым числом параметров за счёт передачи знаний от нейронной сети с большим числом параметров. Где под процедурой передачи знаний понимается обучение нейронной сети воспроизводству обобщений в данных, извлекаемых большей по числу параметров моделью.

Сеть-учитель - в задаче дистилляции обученная нейронная сеть с большим числом обучаемых параметров.

Сеть-ученик - в задаче дистилляции обучаемая нейронная сеть с малым числом обучаемых параметров.

Последний слой нейронной сети - (англ. logit слой) выходной слой архитектуры сети осуществляющий классификацию лиц по входному вектору от извлекателя признаков на заданное число классов. Используется на этапе обучения модели распознавания лиц.

Введение

Сегодня свёрточные нейронные сети повсеместно применяются в задачах автоматизации. Нейросетевые модели позволяют решать широкий спектр задач по анализу изображений, что позволяет широко применять их на практике. К наиболее часто решаемым свёрточными нейронными сетями задачам относятся: классификация изображений [1—3], детектирование объектов на изображениях [4; 5], сегментация изображений [6; 7] и генеративные модели [8—10].

Нейронные сети, впервые описанные в 1943 году Мак-Калоком и Питтсом [11], в 2012 году получили широкое распространение. Толчком к возобновлению интереса к нейросетевым технологиям стала победа нейронной сети AlexNet [12] на конкурсе ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [1], где модель должна была классифицировать изображения на 1000 категорий. Нейронная сеть с архитектурой AlexNet производила классификацию 150'000 изображений с точностью 63%, что на 12% опережало лучшее решение без использования нейронных сетей: SIFT + Fisher Vectors [13]. Высокая точность AlexNet обусловлена использованием свёрточных слоёв, хорошо подходящих для решения задач анализа изображений. Свёрточный слой был предложен в [2] и позволяет извлекать признаки из анализируемого изображения, учитывая их локальную взаимосвязанность.

Современные нейросетевые модели, такие как CoAtNet-7 достигают точности на задаче классификации ImageNet равной 90.9%. Развитие технологий обучения нейронных сетей [14] и разработка новых, более глубоких архитектур [15] позволили для большинства прикладных задач автоматизации получить точности, сравнимые с точностью человека-оператора.

Характерным примером такой задачи автоматизации является задача распознавания лиц. Системы биометрической идентификации по лицу, как правило, состоят из трёх элементов:

1. Нейронная сеть для обнаружения лиц на изображениях. Как правило, такая модель обнаруживает лица в виде ограничивающих прямоугольников, заданных координатами центра и длинами сторон;

2. Нейронная сеть для обнаружения для каждого лица ключевых точек, по которым производится его выравнивание для дальнейшей обработки. Современные детекторы лиц, такие как Retinaface [16] позволяют выполнять одновременное детектирование лица и его ключевых точек, объединяя данный шаг с предыдущим;
3. Нейронная сеть для построения биометрических векторов лиц, измеряя расстояния между которыми можно делать выводы о схожести лиц на изображениях. Такая нейронная сеть преобразует изображение лица в вектор, обрабатывая его последовательностью обучаемых фильтров и применения операторы понижения размерности.

В настоящий момент глубокие нейронные сети позволяют достигать достаточно высокой точности для каждой из этих задач. Так нейросетевой детектор с архитектурой Retinaface позволяет достигать среднего по всем классам значения метрики Average Precision (mAP, описывается в разделе 2.4) более 55% на наборе данных Wider face [17], содержащем сложные для детектирования сценарии. Средний AP демонстрирует среднюю точность при варьировании порога для предсказанных моделью вероятностей во всём диапазоне его значений. Точность в 55% достаточная для применения его в прикладных задачах, требующих обнаружения лица на изображениях. Нейронная сеть с архитектурой ResNet100 [15] достигает точности более 98% для широко применяемого протокола тестирования для задачи идентификации MegaFace [18].

Такие нейросетевые модели позволяют производить идентификацию и верификацию лиц с точностью, достаточной для задач автоматизации. Но их практическая применимость ограничена требованиями к вычислительным ресурсам. Для обработки изображений такими нейросетевыми моделями требуются вычислительные сервера, оборудованные графическими или тензорными процессорами. Так, например, нейронная сеть с архитектурой ResNet100 для построения биометрического вектора тратит 400 мс на восьмиядерном процессоре с частотой 3.50 ГГц. Такого времени работы сети явно недостаточно для решения задачи распознавания лиц в видеопотоке. А оборудование сервера графическим или тензорным процессором приводит к значительному повышению его стоимости.

Для повышения практической применимости нейронных сетей необходимо избавиться от ограничений, вызванных требуемыми вычислительными ресурсами. Одно из решений проблемы высокой вычислительной сложности нейронных сетей — это использование нейросетевых архитектур с меньшим числом обучаемых параметров [19—22]. Также были предложены новые быстрые архитектуры для частных задач, таких как построение биометрических векторов по изображению лиц. К таким архитектурам относятся MobileFaceNet [21] и MobiFace [22]. Такие нейросетевые модели разработаны для применения в маломощных вычислителях, таких как мобильные телефоны, умные домофоны или камеры наружного наблюдения. MobileFaceNet на том же вычислителе обрабатывает изображение за 42 мс против 400 мс для ResNet100. Однако точность таких моделей ниже их серверных аналогов. Разница в точности идентификации на протоколе MegaFace между серверной архитектурой ResNet100 и мобильной архитектурой MobileFaceNet составляет порядка 8%.

Дополнительные сложности с вычислением нейронных сетей на централизованном сервере связаны с вопросом безопасности. Для работы системы распознавания лиц необходимо хранить базу биометрических данных пользователей системы. Таким образом, злоумышленник, получив доступ к серверу, получит доступ ко всей базе пользователей. Выполнение нейронных сетей на конечном устройстве пользователя (мобильном телефоне или домофоне) позволит создать более безопасную распределённую систему. К тому же обработка данных на устройстве пользователя снимает необходимость передачи их на сервер, что требует дополнительных временных затрат и наличия каналов передачи данных.

Подходы к сокращению разрыва в точности между серверными и мобильными архитектурами нейронных сетей можно разделить на четыре категории: прунинг [23—25], дистилляция [26—28], квантование [29—31] и нейросетевой архитектурный поиск [32—34]. Прунинг — это подход к ускорению обученной нейронной сети за счёт удаления весов, оказывающих наименьшее влияние на результат. Дистилляция — это метод обучения нейронной сети с меньшим числом параметров, опираясь на обученную сеть с большим числом параметров. Квантование — это процедура дискретизации весов обученной нейронной сети

для ускорения операций над ними. Нейросетевой архитектурный поиск - позволяет в автоматическом режиме, с учётом домена работы сети и ограничений на её точность и скорость, выбрать оптимальную архитектуру. Однако существующие подходы к оптимизации нейронных сетей не универсальны и требуют разработки отдельных методов для различных классов прикладных задач.

Целью диссертационной работы является разработка методов, позволяющих сократить разницу в точности между серверными и мобильными нейросетевыми архитектурами для задачи распознавания человека по лицу. Предлагаемые методы позволяют улучшить соотношение скорости и точности для нейронных сетей в системе распознавания лиц.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Исследовать эффективность современных методов ускорения нейросетевых моделей и их применимость к нейронным сетям, используемым в задаче распознавания лиц;
2. Разработать метод ускорения обработки потокового видео нейросетевым детектором лиц для встраивания детектора в устройства с маломощными мобильными вычислителями;
3. Разработать метод максимизации скорости работы и минимизации объёма занимаемой памяти для нейронных сетей, выполняющих построение биометрического вектора по изображению лица;
4. Исследовать возможность комбинирования предложенных методов оптимизации и их практическую применимость в задаче идентификации на встраиваемых устройствах, таких как умные домофоны.

Научная новизна:

1. Предложен и реализован новый алгоритм дистилляции для моделей, обученных с функцией Софтмакс с отступом, для задачи построения биометрического вектора по изображению лица. Софтмакс с отступом — это модификация функции Софтмакс с добавлением константы для большего разнесения векторов в пространстве. Предложенный алго-

ритм впервые использует адаптивное вычисление отступов в функции Софтмакс на основе расстояния до центра кластера;

2. Предложен подход, позволяющий эффективнее разносить на гиперсфере биометрические вектора, полученные нейронной сетью с малым числом параметров. Предложенный подход впервые использует модель с большим числом параметров для нахождения центров кластеров векторов для повышения эффективности обучения малой нейронной сети;
3. Разработан и впервые применён метод ранней остановки исполнения нейросетевого детектора объектов на основе значения признаков промежуточных слоёв сети. Продемонстрировано повышение средний скорости обработки кадров за счёт использования предложенного метода;
4. Предложен и реализован устойчивый к шуму алгоритм локализации движения в видеопотоке. Впервые для решения этой задачи применены глубокие признаки предобученного детектора объектов, что не влечёт дополнительных вычислительных затрат.

Теоретическая значимость полученных результатов заключается в предложенном в диссертации новом подходе к передаче знаний от сети-учителя к сети-учинику для задач нейросетевой биоидентификации. Копирование последнего слоя сети-учителя в сеть-ученик позволяет последнему использовать более хорошее пространство векторов при обучении. Предлагаемый метод позволяет расширить пространство возможных каналов передачи знаний между нейронными сетями при дистилляции.

Практическая значимость полученных результатов заключается в реализации разработанного подхода в виде программного комплекса для обучения свёрточных нейронных сетей на ЭВМ. Получаемые таким образом нейросетевые модели могут использоваться для распознавания лиц на маломощных вычислителях. Описанный в данной диссертации подход позволил реализовать алгоритмы обнаружения и идентификации лица в видеопотоке на конечном устройстве пользователя с мобильным ARM процессором. Технология распозна-

вания лиц, основывающаяся на разработанном подходе внедрена в продуктах компаний:

1. “ООО Новотелеком”. Нейронные сети, обученные с помощью разработанного подхода, применяются в Проекте «Свободные руки»: системе идентификации жильцов по видео с камеры домофона;
2. “ООО Рубетек РУС”. Разработанный подход применялся для обучения нейронной сети для распознавания людей, имеющих доступ в помещение, по камере, установленной на входе. Данная технология применяется в жилых комплексах Группы Компаний ПИК;
3. “ООО Открытая мобильная платформа”. Обученные с использованием разработанного метода нейронные сети применяются для распознавания лиц в операционной системе “Аврора”.

Основные положения, выносимые на защиту:

1. Использование обучаемых весов последнего слоя сети-учителя для инициализации весов сети-ученика в задаче дистилляции СНС для построения биометрических векторов лиц позволяет получать более компактные кластеры биометрических векторов на гиперсфере;
2. Инициализация весов копированием последнего слоя из большей модели вынуждает сеть-ученик генерировать вектора с сохранением пространственных взаимоотношений как в сети-учителе, что позволяет использовать модели разной ёмкости в одной системе и открывает новые возможности для разработки методов дистилляции нейронных сетей;
3. Реализация предложенного подхода инициализации весов последнего слоя нейронной сети продемонстрировала повышение процента верно верифицированных изображений лиц на открытом наборе данных;
4. При использовании Софтмакс с отступом, вычисление отступа на основе угла между биометрическим вектором и центром класса, вычисленных нейронной сетью с большим числом параметров, позволяет увеличивать точность моделей для построения биометрических векторов лиц. Софтмакс с отступом — это модификация функции Софтмакс

с добавлением константы для большего разнесения векторов в пространстве;

5. Реализация предложенного подхода дистилляции на основе адаптивного отступа в Софтмакс продемонстрировала повышение процентов верно идентифицированных и верифицированных людей на открытых наборах данных;
6. Использование карт активаций на промежуточных слоях извлекателя признаков в задаче детектирования позволяет производить устойчивое к зашумлённости входного видеопотока обнаружение движения. Применение такой техники обнаружения движения в кадре позволяет с высокой надёжностью отфильтровывать статичные кадры из зашумлённого видеопотока и не требует дополнительных вычислительных ресурсов;
7. Реализация предложенного подхода был экспериментально проверена на открытых наборах данных и продемонстрирована его эффективность в сравнении с подходом на основе попиксельного сравнения кадров;
8. Разработанный комплекс программных средств для обучения СНС через дистилляцию позволяет создавать программные решения на основе нейросетевых технологий для задачи распознавания лиц, работающие на борту встраиваемого устройства или маломощном сервере;

Достоверность полученных результатов обеспечивается корректным проведением большого числа тестов на реальных данных, в том числе независимым институтом стандартов. Для замера точности системы использовались различные объективные метрики, продемонстрировавшие результаты непротиворечивые друг-другу и теоретическим выкладкам.

Апробация работы. Основные положения и результаты диссертационной работы докладывались на конференциях: “ICDLFR 2020: 22nd International Conference on Deep Learning and Face Recognition”, Амстердам, 2020 г.; “ММРО 2021: Математические методы распознавания образов”, Москва 2021 г. Доклад на конференции ICDLFR 2021 был отмечен дипломом за лучший доклад.

Личный вклад. Автор разработал и реализовал подходы к оптимизации производительности свёрточных нейронных сетей в системе распознавания лиц. Разработанные подходы описаны в главах: "Глава 2 Оптимизация детектора лиц"и "Глава 3 Оптимизация вычисления биометрического вектора". Автор провёл полное тестирование разработанной системы как описано в "Глава 4 Разработка программного модуля для встраиваемых систем".

Публикации. Материалы диссертации опубликованы в 5 печатных работах в рецензируемых журналах [28; 35—38]. Работы [35; 36] в индексируемых Scopus журналах, относящихся к квартили Q1. Работы [37; 38] изданы в рецензируемых журналах, индексируемых ВАК РФ и относящихся к квартилям Q3 и Q2.

Объем и структура работы. Диссертация состоит из введения, 4 глав и заключения. Полный объём диссертации составляет 109 страниц, включая 23 рисунка и 18 таблиц. Список литературы содержит 106 наименований.

Первая глава носит обзорный характер и содержит результаты исследования современных методов оптимизации производительности нейронных сетей. В ней рассмотрены следующие категории подходов: дистилляция, квантование, прунинг, нейросетевой архитектурный поиск. На основе литературных источников проведён анализ существующих подходов с целью выявления их сильных сторон и недостатков, а также анализ применимости различных алгоритмов к задаче биоидентификации человека по лицу. Выявлены узкие места и проблемы в процедуре применения подходов для ускорения нейронных сетей к системе распознавания лиц, на решении которых сосредотачивается данная диссертация.

Во второй главе предлагается подход к ускорению алгоритма обнаружения человека в видеопотоке за счёт снижения числа ложноположительных детекций. Ускорение достигается путём снижения количества вычисляемых биометрических векторов, а также за счёт предложенного метода ранней остановки работы детектора для статичных кадров. Предлагаемый подход применим к большинству современных систем детектирования объектов. В главе экспериментально показывается увеличение средней скорости обработки кадра и повышение точности работы детектора.

В третьей главе описывается предлагаемый метод для дистилляции сетей, обученных со специализированной функцией ошибки для систем распознавания лиц. Предлагаемый подход учитывает особенности работы таких сетей в процессе их дистилляции, что позволяет получать большую точность в сравнении с другими современными методами дистилляции. Экспериментально показывается увеличение точности модели в сравнении с сетями, полученными с помощью других подходов на наиболее распространённых наборах тестовых данных, используемых в данной области.

В четвёртой главе приводится описание деталей реализации подхода на языках программирования Python и C++, особенности обучения нейронных сетей и внедрения предложенных в данной диссертации подходов. Также в четвёртой главе приводятся результаты обширного тестирования разработанной системы распознавания лиц на реальных данных, в том числе независимым национальным институтом стандартов.

Глава 1. Методы оптимизации скорости работы нейронных сетей

В настоящей главе приведён обзор на основе литературных источников методов оптимизации скорости работы нейросетевых моделей и применимость этих методов к технологии распознавания лиц. Рассмотрены принципы работы и основные характеристики таких методов как прунинг, дистилляция, квантование и нейросетевой архитектурный поиск.

1.1 Классификация методов оптимизации скорости нейросетевых моделей

С развитием и повсеместным распространением мобильных CPU остро стал вопрос об ускорении нейронных сетей для портирования на маломощные вычислители. Мобильные CPU обладают меньшей вычислительной мощностью, чем их аналоги для настольных компьютеров и серверов. Такое снижение вычислительной мощности CPU осуществляется в пользу повышения их энергоэффективности и компактности. Так средняя мощность мобильного CPU составляет порядка 100 Гига-FLOPS, а CPU для персонального компьютера 1000 Гига-FLOPS. Сравнение мощности мобильных CPU и CPU для ПК представлена на диаграмме (Рис. 1.1).

Также для обработки нейронных сетей в серверах и ПК применяются вычислители с массивно параллельной архитектурой - GPU. В последнее время набирает популярность применение специализированных тензорных вычислителей - NPU. Однако их мобильные аналоги также значительно уступают в вычислительной мощности.

Такой разрыв в производительности между мобильными вычислителями и настольными ПК побудил научное сообщество к разработке методов ускорения нейронных сетей с минимальными просадками точности. Методы оптимизации нейронных сетей можно разделить на следующие четыре категории:

– Прунинг;
– Дистилляция;
– Квантование;
– Нейросетевой архитектурный поиск.

К первой категории “Прунинг” относятся методы, основанные на удалении из нейронной сети весов, наименьшим образом влияющих на результат работы сети. Выбор весов производится в соответствии с некоторым критерием их важности для итогового результата и на каждом шаге прунинга удаляется некоторый процент самых не важных весов с последующим дообучением сети.

К категории “Дистилляция” относятся подходы к обучению нейронной сети с меньшим числом параметров за счёт передачи знаний от сети с большим числом параметров. Ключевая идея, лежащая в основе такого подхода, заключается в том, что нейронную сеть малой ёмкости можно обучить для достижения большей точности за счёт передачи информации от обученной сети большей ёмкости. Где под передачей знаний понимается обучение малой сети повторять некоторые свойства большой, такие как параметры распределений весов и расстояния между выходными векторами.

“Квантование” объединяет методы понижения дискретности весов нейронной сети для более эффективной утилизации вычислителя. Так нейронная сеть, обученная с использованием 32-битных чисел с плавающей запятой, может быть заквантована и выполняться на мобильном процессоре в 8-битных беззнаковых целых числах. Подходы к квантованию основываются на выборе коэффициентов сдвига и масштабирования для перевода значений переменной в более узкий диапазон.

К последней категории “Нейросетевой архитектурный поиск” относятся методы, позволяющие не ускорить уже обученную сеть, а в автоматическом режиме найти оптимальную нейросетевую архитектуру для решаемой задачи: для некоторых обучающих данных и целевого вычислителя. Как правило, методы из данной категории принимают на вход некоторый набор блоков поиска - изолированных частей нейросетевой архитектуры. Затем алгоритм комбинирует их оптимальным образом с учётом желаемых характеристик получаемой нейронной сети.

Подходы из представленных выше категорий можно комбинировать между собой. Так, в начале применив нейросетевой архитектурный поиск, можно выбрать наиболее оптимальную архитектуру, а затем обучить её, используя дистилляцию. К полученной таким образом модели применить алгоритм прунинга, чтобы исключить веса нейронной сети, наименьшим образом влияющие на точность. И итоговую модель заквантовать для выполнения на мобильном вычислителе. Далее в этой главе подробно рассматривается каждая из представленных категорий.

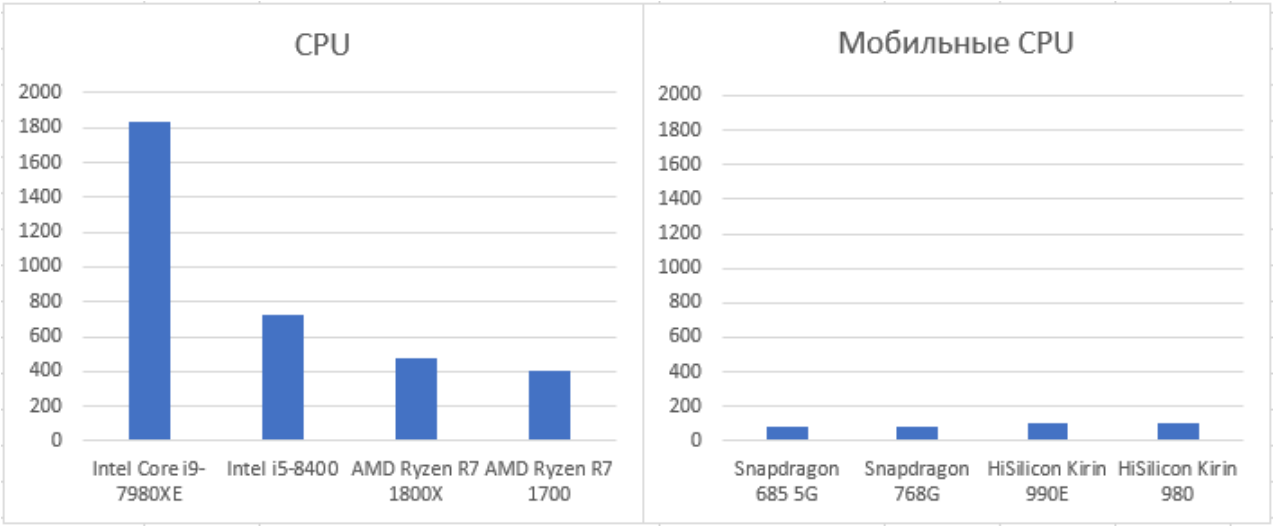

Рисунок 1.1 — Показатели Гига-FLOPS для различных моделей CPU для настольных ПК и мобильных CPU.

1.2 Прунинг

Прунинг является одним из наиболее старых способов ускорения нейронных сетей и применяется для чрезмерно параметризованных моделей. В основе прунинга лежит идея об удалении связей между нейронами, вносящих наименьший вклад в итоговый результат. На практике под удалением часто понимается зануление соответствующих весов нейронной сети.

Первыми, кто занялся изучением удаления связей между нейронами как способа ускорения и сжатия нейронных сетей, были авторы вышедшей в 1989

году статьи “Optimal brain damage” [39]. Предлагаемый авторами данной работы подход к прунингу можно описать следующим алгоритмом:

1. Обучить нейронную сеть;
2. Измерить важность каждого обучаемого параметра нейронной сети, измеряя отклонения функции потерь при возмущениях в значении каждого параметра;
3. Удалить из нейронной сети параметры, вызывающие наименьшие отклонения функции потерь;
4. Дообучить полученную нейронную сеть, чтобы оставшиеся параметры могли скорректировать свои значения с поправкой на изменения в модели.

Описанный алгоритм можно выполнять итеративно, на каждом шаге удаляя некоторый небольшой процент обучаемых параметров нейронной сети. В таком случае оставшимся параметрам на каждой итерации будет проще подстраиваться.

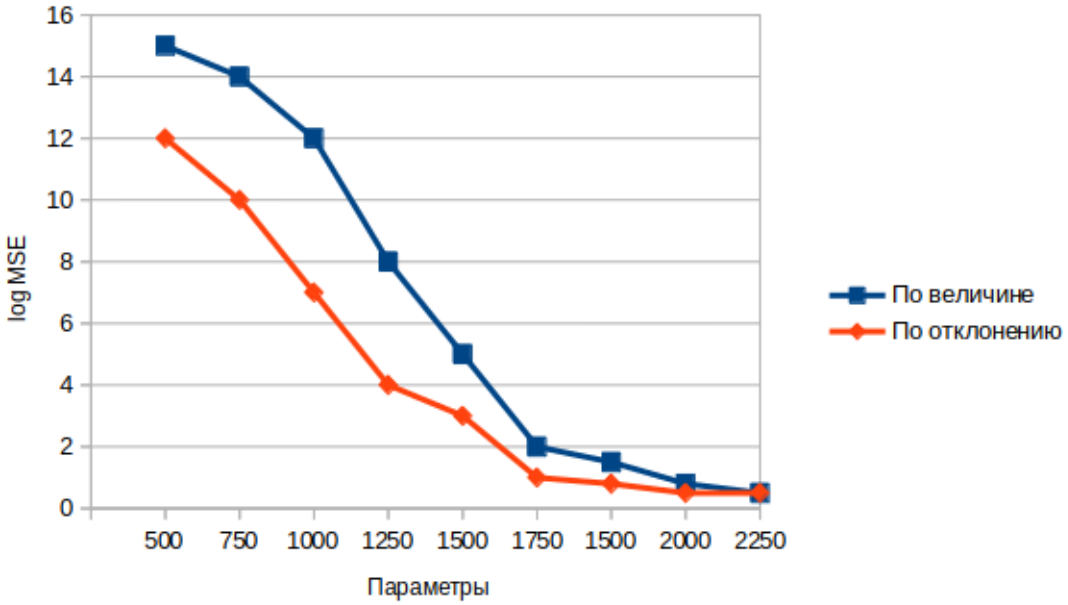

Рисунок 1.2 — Графики из работы [39]. Показывает отношение функции ошибки (по вертикали) к числу параметров нейронной сети (по горизонтали) при прунинге. Верхняя кривая отражает значения при прунинге по абсолютному значению весов; Нижняя кривая - значения при прунинге по отклонению целевой функции.

В недавнее время алгоритмы прунинга получили новый виток развития, новые подходы позволяют производить прунинг нейронных сетей с меньшей просадкой точности. Исследователи из MIT в 2018 году предложили альтернативный подход [40] к прунингу нейронных сетей основанный на “гипотезе о лотерейных билетах”.

Гипотеза о лотерейных билетах заключается в следующем: если случайным образом инициализировать веса нейронной сети, то найдётся такая её подсеть, которая за меньшее или, в худшем случае, такое же число шагов алгоритма обучения, будет сходиться к той же точности, что и исходная сеть.

Интуитивно данная гипотеза объясняется тем, что мы можем выбрать подсеть достаточно большим количеством способов, чтобы утверждать, что вероятность её существования не нулевая. Однако выбор такой подсети аналогичен вытягиванию выигрышного билета в лотерее, что и привело к такому названию гипотезы. В статье же авторы предлагают алгоритм поиска такой подсети:

1. Параметры нейронной сети инициализируются случайными значениями из некоторого распределения. Значения проинициализированных весов сохраняются в специальный буфер;
2. Нейронная сеть обучается до сходимости;
3. Выбирается желаемый процент параметров для удаления из сети на основе их абсолютного значения;
4. Нейронная сеть обучается без этих параметров. Все остальные параметры инициализируются значениями, вычисленными на шаге 1, которые берутся из буфера.

Предложенный алгоритм также выполняется итеративно, удаляя на каждом шаге небольшой процент параметров модели до тех пор, пока мы не достигнем желаемого размера сети или не получим критическое снижение точности. Полученная таким образом сеть будет являться подсетью исходной сети и для зафиксированной случайной инициализации будет достигать точности сравнимой с исходной моделью.

Хотя такой метод прунинга хорошо показал себя на простых архитектурах и наборах данных, таких как Lenet и Mnist, как можно видеть на Рис. 1.3,

для более сложных архитектур, таких как ResNet-20, требуется дополнительная настройка шага обучения, иначе рассматриваемый метод не даёт преимущества в сравнении со случайной инициализацией весов на каждом шаге прунинга. Из графика Рис. 1.3(б) можно видеть, что выигрыш по точности достигается только при снижении шага обучения и использовании техники прогрева весов.

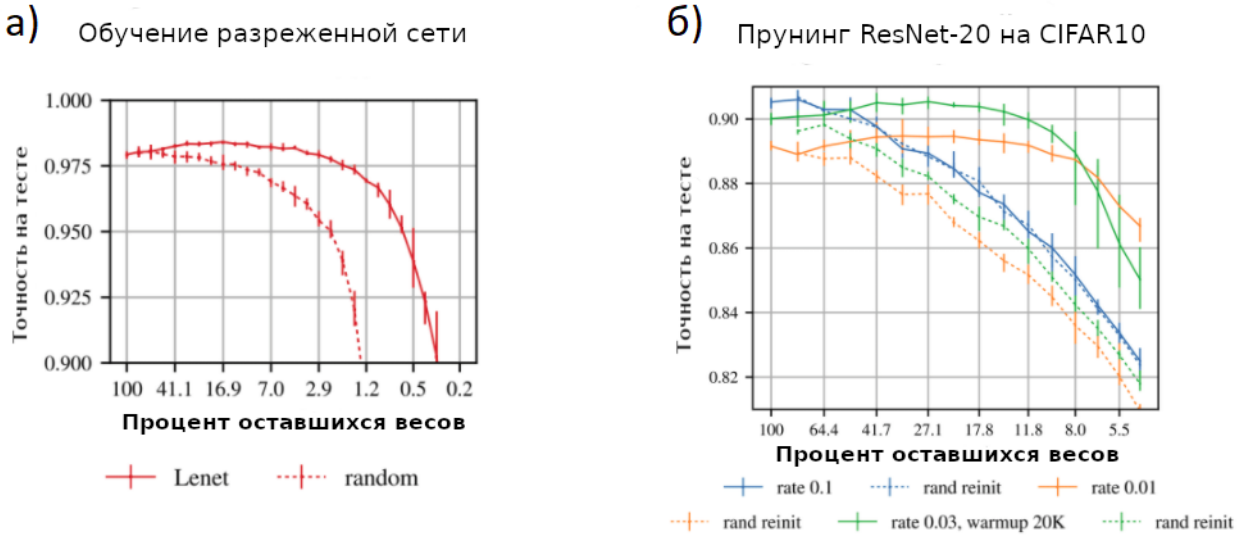

Рисунок 1.3 — Графики из работы [39]. (а) Сравнение точности модели для различного числа параметров в сети при случайной инициализации (пунктирная линия) параметров после каждого шага прунинга и при инициализации буферным значением (непрерывная линия). (б) Сравнение точности модели с архитектурой ResNet-20 для прунинга различного процента весов.

Авторы преодолели ограничение метода прунинга основанного на “гипотезе о лотерейном билете“ в своей следующей работе [41]. Для того, чтобы предложенный метод мог работать для сложных наборов данных, таких как ImageNet [1], и архитектур, таких как ResNet [15], без изменения шага обучения, авторы предложили ослабить ограничения для “выигрышного билета” - подархитектуры, достигающей той же точности, что и исходная. В качестве такого ослабления в новой работе предлагается на каждой итерации прунинга сбрасывать веса не до начальной инициализации, а до некоторого шага обучения сети. Таким образом находимая архитектура будет не “выигрышным билетом”, а совпавшим с предыдущей итерацией билетом. На рисунке 1.4 видно, что такая модификация позволяет достичь меньшей просадки точности при прунинге в

сравнении с оригинальной “гипотезой о выигрышном билете” и наивным прунингом со случайной инициализацией параметров сети после каждой итерации.

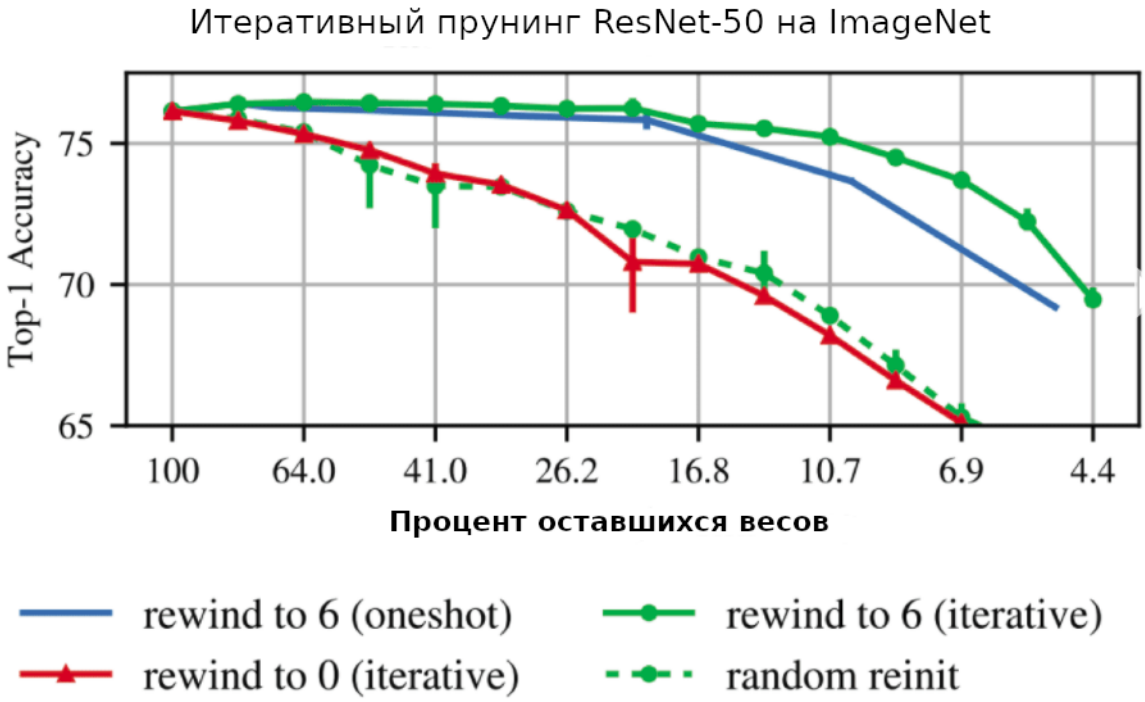

Рисунок 1.4 — Графики из работы [41]. Сравнение снижения точности для методов прунинга на основе “гипотезы о лотерейных билетах” для инициализации параметров начальными значениями и с заданного шага.

Среди недавних алгоритмов прунинга можно выделить SynFlow [42]. Авторы данной работы предложили подход к прунингу параметров нейронной сети не требующий её переобучения. Другими словами, авторами предлагается выбирать “выигрышный лотерейный билет” на этапе инициализации сети. Авторы вводят понятие коллапса слоя - прунинга всех обучаемых параметров в слое, что делает сеть не пригодной для обучения. Затем формулируют алгоритм позволяющий избегать таких коллапсов. Суть алгоритма заключается в измерении для каждого обучаемого параметра некоторого показателя, равного произведению значения параметра на производную функции ошибки по нему. Параметры для прунинга при этом выбираются таким образом, чтобы суммы входящих и выходящих в нейрон показателей совпадали. Такой подход позволяет избегать коллапса слоёв.

В целом существующие на сегодняшний день методы прунинга можно разделить по следующим признакам:

- Структура прунинга: структурированный прунинг и неструктурированный прунинг. В первом случае прунинг параметров сети выполняется группами с удалением нейронов целиком. Неструктурированный прунинг, в свою очередь, выполняется без учёта связи между параметрами сети. Хотя неструктурированный прунинг позволяет удалять большее количество параметров, вычисление сети после такого прунинга, как правило, менее эффективно, чем после структурированного. Это связано с неэффективностью разреженных вычислений на вычислителе.
- Способ выбора параметров для прунинга: прунинг на основе величины веса или прунинг на основе величины проникающего через него градиента. Первый способ может быть интуитивно не понятен, так как при обучении сети используются различные техники регуляризации для снижения величины весов, поэтому всё больше исследователей прибегают ко второму способу.
- Частота прунинга: итеративно или один раз в конце процедуры обучения. Практика показывает, что итеративный прунинг небольшого числа параметров с последующим дообучением сети позволяет получить меньшее снижение точности.

Алгоритмы прунинга реализованы в таких популярных библиотеках для глубокого машинного обучения как Tensorflow [43] и Pytorch [44]. Библиотека Tensorflow использует алгоритм прунинга в своём API, реализуя его как итеративное удаление параметров сети, основываясь на их абсолютном значении. Tensorlow API позволяет регулировать процент удаляемых весов, как часто применять алгоритм прунинга в процессе обучения и структурированность прунинга - удалять параметры по одному или блоками.

API библиотеки Pytorch включает множество методов структурированного и неструктурированного прунинга. Также Pytorch предоставляет удобный интерфейс для реализации своих методов прунинга, что даёт большую гибкость при использовании данного фреймворка.

Хотя алгоритмы прунинга хорошо себя показывают в задачах оптимизации производительности нейронных сетей за счёт сокращения числа обучаемых параметров, их применимость ограничена только избыточно параметризованными архитектурами. Наиболее эффективны алгоритмы прунинга в задачах, когда необходимо оптимизировать сеть с избыточно большим числом параметров для сравнительно простого набора данных. Такого набора данных, который не обладает высоким разнообразием классов объектов и содержит легко отличимые СНС паттерны. На сегодняшний день, с развитием методов автоматического нейросетевого поиска архитектур сетей, задача оптимизации излишне параметризованных сетей встречается гораздо реже. При решении прикладной задачи разработчик может выбрать достаточно эффективную архитектуру в автоматическом режиме.

1.3 Дистилляция

Дистилляция — это способ более эффективно обучить модель с малым числом параметров, используя информацию, получаемую от обученной модели с большим числом параметром. Изначально идею дистилляции как передачи информации между моделями для их обучения предложил Хинтон в своей работе [26]. В ней вводится понятие “dark knowledge” (тайное знание). В данном случае под тайным знанием Хинтон подразумевает способность сети обобщать данные и извлекать из них определённые признаки, такие, что сеть с малым числом параметров способна их извлечь, но не способна самостоятельно подобрать для этого значения параметров. Утверждается, что если мы будем обучать сеть с меньшим и большим числом параметров, то при условии, что меньшая сеть может показывать точность равную большей сети, она может быть не способна самостоятельно обучиться до такой точности. При том, что малая сеть имеет достаточное количество обучаемых фильтров для извлечения признаков, её ёмкости может быть недостаточно для того, чтобы самостоятельно выучить веса фильтров. Вводимое Хинтоном “тайное знание” позволяет

передавать дополнительную информацию от уже обученной сети, чтобы сеть с меньшим количеством параметров смогла настроить значения весов фильтров. Большая сеть, используемая для извлечения “тайного знания”, называется сетью учителем. Меньшая сеть, принимающая “тайное знание”, называется сетью учеником. Сеть-учитель, как правило, значительно больше сети-ученика и может представлять собой даже ансамбль нейронных сетей.

В оригинальной работе по дистилляции [26] в качестве “тайного знания” предлагается использовать распределение вероятностей для предсказаний нейронной сети. В случае решения задачи классификации на выходе нейронной сети будет получаться вектор, удовлетворяющий определению вероятности: каждое его значение показывает уверенность сети в принадлежности входных данных к классу с определённым номером; сумма элементов вектора равняется единице. Для передачи знаний используется сглаженное распределение вероятности, для этого в функцию Софтмакс (рассматривается в разделе 3.1) на последнем слое сети Хинтон предлагает вводить температуру T - все входные значения функции Софтмакс делятся на T:

$$q_i = \frac{e^{z_i/T}}{\sum_k e^{z_k/T}}$$

Такое сглаживание позволяет лучше передавать информацию о распределении уверенности сети между различными целевыми классами. Например, изображение 1.5 модель с вероятностью более 0.8 относит к классу "Английский спрингер спаниель". Но при введении температуры в Софтмакс можно видеть (Рис. 1.6) к каким классам с меньшей вероятностью СНС отнесла изображение: уэльский спрингер спаниель, английский кокер спаниель и т.д.. Такое сглаженное распределение вероятностей даёт информацию о логике работы модели. Сеть-ученик обучается, используя расхождение Кульбака-Лейблера между распределениями сетей ученика и учителя в качестве функции ошибки. Подражание распределению сети-учителя позволяет сети-ученику выучивать значения параметров для достижения большей точности.

Описанный подход также применим при использовании ансамбля сетей в качестве сети-учителя. В работе [45] предлагается производить дистилляцию из

Рисунок 1.5 — Пример изображения которое нейронная сеть распознаёт как спаниеля.

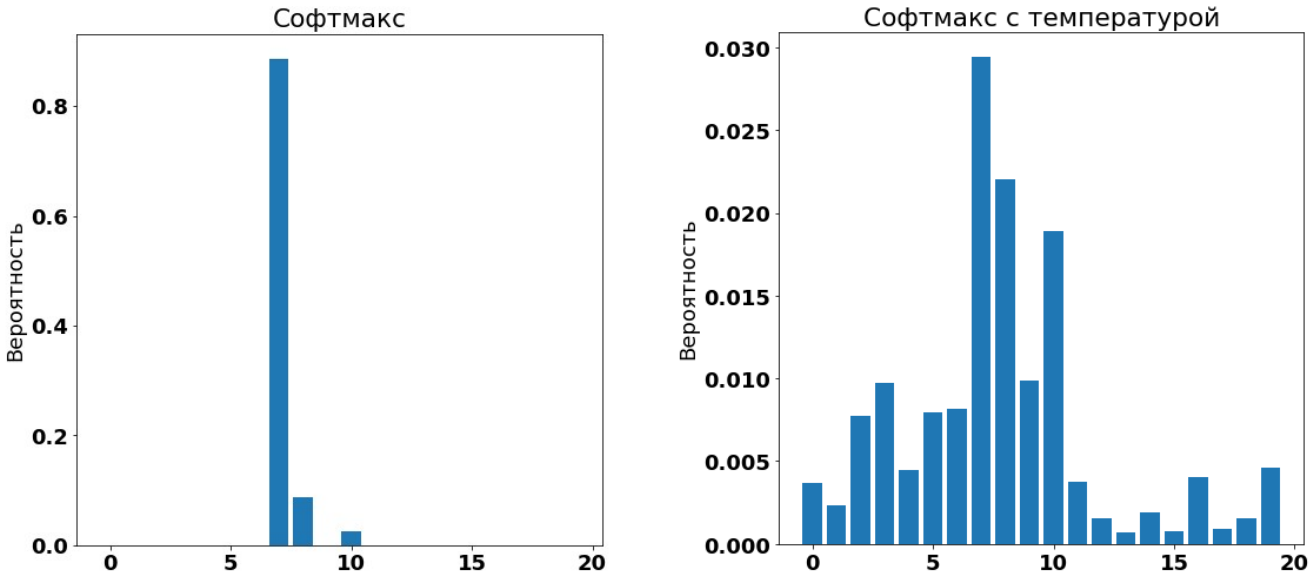

Рисунок 1.6 — Распределение предсказанных моделью вероятностей между классами для Софтмакс и Софтмакс с температурой.

нескольких VGG или LSTM сетей для повышения точности в задаче распознавания речи. Авторы приводят два новых подхода к использованию информации от нескольких сетей для обучения сети-ученика помимо использования взвешенной суммы предсказаний сетей в ансамбле:

– Переключение учителя: на каждом шаге обучения в качестве целевого предсказания для ученика выбирается ответ случайного учителя;

– Использование аугментаций: каждый учитель получает на вход различным образом аугментированные данные. Для обучения ученика используются ответы всех учителей.

В [46] предлагается осуществлять передачу “тайного знания” решая задачу регрессии для предпоследних слоёв сети, содержащих логарифмические вероятности, так называемых logit-слоёв. Для этого в качестве функции ошибки вводится L_2 расстояние между logit-слоями:

$$L_2(x, z, \theta) = \frac{1}{2T} \sum_i \left\| g(x^{(i)}; \theta) - z^{(i)} \right\|_2^2,$$

где T обозначает размер пакета изображений, $x^{(i)}$ означает i-й тренировочный пример в пакете, $z^{(i)}$- соответствующее значение logit-слоя сети-учителя, θ - параметры сети-ученика, $g(x^{(i)}; \theta)$ - значение logit-слоя сети-ученика.

Работа [47] про зашумлённого учителя развивает эту идею. В основе данной работы лежит идея, что при дистилляции из ансамбля моделей получается достичь большей точности сети-ученика, но при отсутствии реального ансамбля его можно симулировать. Для симуляции ансамбля предлагается добавлять к logit-слою сети-учителя некоторый шум. Такое добавление шума позволяет не только симулировать дистилляцию из ансамбля сетей, но и оказывает регуляризирующий эффект, предотвращая переобучение сети-ученика.

Обучение сети-ученика с применением техник дистилляции в некоторых случаях [48] позволяет даже превзойти точность сети-учителя. Применяя технику дистилляции, авторы данной работы смогли получить наибольшую точность для наборов данных CIFAR-10 и CIFAR-100. Это достигается за счёт использования техники дистилляции не для сокращения числа параметров в сети, а для повышения её обобщающей способности. Для этого авторы обучают последовательно несколько нейронных сетей с одинаковой архитектурой и случайной инициализацией параметров, используя в качестве обучающих данных предсказания предыдущей в цепочке обучений сети. Итоговую последовательность сетей можно объединить в ансамбль для большего увеличения точности модели.

В недавних работах предлагается использовать для дистилляции знаний не только значения получаемые на последних слоях сети, но и значения с промежуточных слоёв. Так в [49] предлагается использовать дистилляцию на

Таблица 1 — Сравнение точности сети ученика и учителя для подхода дистилляции, описанного в [49].

Алгоритм	Число параметров	Точность
FitNet	∼2.5 M	64.96%
Teacher	∼9 M	63.54%

промежуточных слоях для увеличения глубины возможного ученика. Для этого авторы вводят дополнительные регуляризирующие члены, основывающиеся на информации от промежуточных слоёв сети-учителя, в функцию ошибки сети-ученика. Задача данных членов в том, чтобы следить чтобы карты признаков выбранных промежуточных слоёв сети-ученика не отклонялись от значений слоёв сети-учителя. Особенностью такого подхода является тот факт, что в качестве сети-ученика может выбираться более глубокая сеть, но с меньшим количеством параметров в каждом отдельно рассматриваемом слое. Что в совокупности с обучением с Хинтоновской дистилляцией с температурой и кросс-энтропией с реальными значениями позволяет достичь точность превосходящую точность сети-учителя (Таблица 1).

Однако такой подход требует использования дополнительных полносвязных слоёв для сравнения значений на выходе промежуточных слоёв сетей ученика и учителя, так как они имеют различное количество параметров, что приводит к разрастанию числа обучаемых параметров. Авторы работ [50] и [51] предлагают схожие подходы к дистилляции с использованием промежуточных слоёв, не требующих ввода дополнительных параметров. Для этого вводится дополнительная функция потерь, штрафующая сеть-ученика, если взаимные расстояния между выходными признаками слоёв для различных тренировочных примеров отличаются от аналогичных расстояний для этих же примеров в сети-учителе.

Несмотря на общность доступных методов дистилляции, для некоторых частных задач специализированные подходы позволяют добиться большей точности. Так в задаче распознавания лиц существует несколько таких подходов. Это обусловлено тем, что наибольшую точность для биометрии по лицу позволяет получить обучение с использованием функции Софтмакс с отступом

(рассматривается в разделе 3.4), такой как CosFace [52], SphereFace [53] или ArcFace [54]. Нейронная сеть, обученная таким образом, производит вектора, соответствующие точкам на поверхности гиперсферы. Близость между такими векторами оценивается по углу, а наибольшую точность при дистилляции позволяют получить подходы, учитывающие эту особенность работы сети.

К таким методам дистилляции относятся триплетная дистилляция [55], дистилляция по углу [56] и дистилляция на основе отступа [57], а также метод, предложенный в рамках данного диссертационного исследования [28]. Последний будет подробно рассмотрен в третьей главе.

Метод триплетной дистилляции [55] основывается на применении триплетной функции ошибки:

$$Loss = \frac{1}{N}\sum_{i=0}^{N} max(D(x_i^a, x_i^p) - D(x_i^a, x_i^n) + m, 0)$$

где D - это расстояние между изображениями в векторном пространстве (угол между точками на гиперсфере); x^a, x^p, x^n - триплет изображений: x^a - якорный или опорный пример, x^p - позитивный пример (относящийся к тому же классу, что якорный), x^n - негативный пример (относящийся к другому классу). В этой формуле m — это гиперпараметр, регулирующий отступ между позитивным и негативным примером. Для осуществления передачи знаний от сети-учителя в методе триплетной дистилляции предлагается вычислять m адаптивным образом, основываясь на разнице расстояний между позитивным и опорным, и негативным и опорным векторами сети-учителя.

Метод дистилляции по углу [56] демонстрирует эффективность явного метода дистилляции, при котором нейронная сеть-ученик обучается таким образом, чтобы угол между некоторыми векторами сети-ученика совпадал с углом между соответствующими векторами сети-учителя. Обучение сети-ученика с явным требованием на совпадения соотношения углов между векторами позволяет повысить точность сети-ученика в сравнении с обучением без дистилляции. Хотя такой подход может страдать от излишней регуляризованности сети и не давать ученику раскрыть весь его потенциал.

Дистилляция на основе отступа [57] наследует идею предложенную Хинтоном и заключающуюся в добавлении параметра температуры для того,

чтобы сгладить распределения на выходе сети. В работе [57] предлагается использовать параметр в формуле для ArcFace по аналогии с классическим использованием данного параметра в функции Софтмакс.

Численное сравнение описанных специализированных методов дистилляции для биометрических нейронных сетей приводится в третьей главе данной диссертации.

1.4 Квантование

Широкое распространение получил такой способ оптимизации производительности и потребляемой памяти нейронной сети как квантование [58]. Под квантованием модели понимается понижение дискретности представления её параметров (Рис. 1.7) с целью более эффективной их обработки на вычислителе и хранения в памяти. Так, например, модель, обученную с параметрами представленными числами с плавающей точкой (float32), можно для исполнения заменить моделью с параметрами представленными целочисленными однобайтовыми переменными. Таким образом сократив объём занимаемой моделью памяти с 32 бит на переменную до 8 бит на переменную.

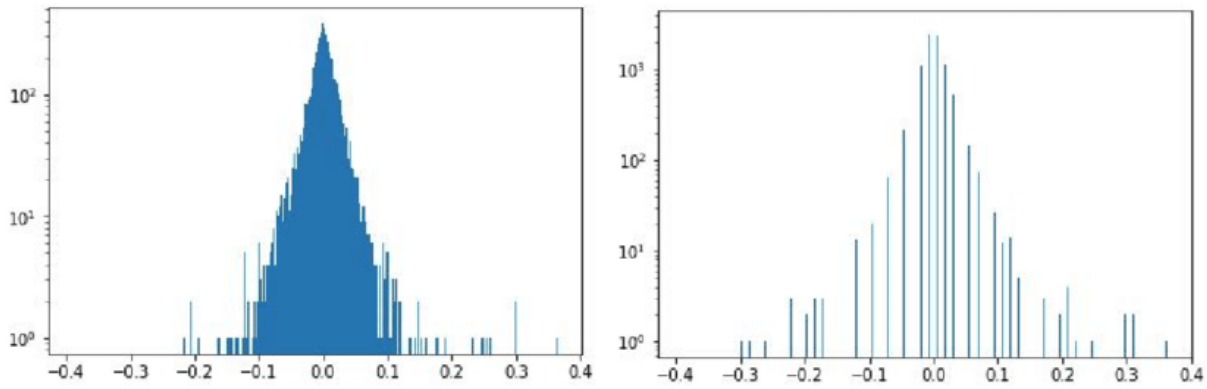

Рисунок 1.7 — Изображение из статьи [59] демонстрирующее различие между распределениями значений параметров для исходной нейронной сети (слева) и квантованной нейронной сети (справа).

Идея квантования параметров математической модели берёт своё вдохновение в нейробиологии, а именно в предположении, что человеческий мозг хранит информацию в квантованном виде. Данное предположение вызвано тем, что дискретная информация меньше подвержена воздействию шума при хранении и обработке [60]. История практического применения квантования восходит корнями к 1800-м годам, когда квантование было впервые применено для аппроксимации вычисления интеграла [61]. Но несмотря на долгую историю развития методов квантования, применение их для ускорения вычисления нейронных сетей ставит перед исследователями новые задачи, такие как поиск оптимального соотношения степени сжатия и точности вычислительно интенсивной модели с миллионами параметров.

Общую формулу квантования параметров нейронной сети можно представить в следующем виде:

$$Q(r) = Int(r/S) - Z,$$

где r – это значение параметра нейронной сети, а S и Z соответственно параметры масштабирования и сдвига. Где параметр масштабирования вычисляется следующим образом:

$$S = \frac{\beta - \alpha}{2^b - 1},$$

где β и α задают верхнюю и нижнюю границу квантования, а b – число бит, в которое производится квантование. В зависимости от значений этих параметров и способов их выбора, методы квантования можно классифицировать по следующим характеристикам:

– Симметричное или асимметричное квантование;
– Статическое или динамическое квантование;
– Поканальное или послойное квантование;
– Квантование при обучении или квантование после обучения.

Симметричное или ассиметричное квантование

При асимметричном квантовании значения $\beta = r_{max}$ и $\alpha = r_{min}$. Такой подход называется ассиметричным потому, что границы квантования не

обязательно симметричны относительно 0: $-\alpha \neq \beta$. Под симметричным квантованием понимается такая политика выбора границ, при которой $-\alpha = \beta$, например $-\alpha = \beta = max(r_{max}, r_{min})$. Также для симметричного квантования значение параметра Z выбирается равным 0, что упрощает работу с ним. Однако выбор ассиметричного квантования позволяет более плотно описать множество значений переменной и таким образом производить вычисления с большей точностью. В нейронных сетях выбор ассиметричного квантования особенно полезен при использовании несимметричных функций активации, таких как ReLU.

Статическое или динамическое квантование

Другой отличительной характеристикой методов квантования является то, когда именно производится выбор порогов квантования β и α. Для обучаемых параметров модели значения порогов можно выбрать фиксированными на этапе обучения. Но так как при вычислении квантованной нейронной сети промежуточные вычисления, такие как карты активаций (выходные значения слоёв активации), обрабатываются в квантованном виде – для них также необходимо выбирать пороги квантования. Такие значения зависят от входных данных при вычислении нейронной сети и выбор порогов квантования может производиться для них статическим либо динамическим способом.

При динамическом квантовании пороги выбираются на этапе выполнения сети, оцениванием максимальных и минимальных значений получаемых величин. Такой подход позволяет получать меньшее снижение точности в сравнении с неквантованной сетью за счёт более точной подгонки диапазонов квантования под каждый пример входных данных. Но вычислительно более затратный, так как требует дополнительных вычислений для оценки порогов.

Альтернативой ему является вычисление порогов на этапе обучения нейронной сети – статическое квантование. Такой подход не требует дополнительных вычислений при выполнении нейронной сети, но, как правило, позволяет получить более низкую точность в сравнении с динамическим квантованием. Для оценки порогов квантования в таком подходе используется их вычисление по некоторому набору калибровочных данных сразу после обучения нейронной сети. Несмотря на снижение точности в сравнении с динамическим

методом квантования, на практике обычно прибегают именно к статичному методу из-за его большей вычислительной эффективности.

Поканальное или послойное квантование

Также пороги квантования β и α в свёрточной нейронной сети можно выбирать как для каждого свёрточного фильтра отдельно, так и для всего слоя целиком. Подход, когда пороги квантования выбираются для слоя целиком, называется послойным квантованием. Хотя такой подход гораздо легче реализуется, он может приводить к снижению точности ввиду отличающихся максимальных и минимальных значений в каждом канале свёрточного слоя. Таким образом точность работы слоя сети может существенно просесть, если он содержит фильтр со значением параметров со значительно более широким диапазоном значений.

Решением данной проблемы является поканальное квантование фильтров свёрточного слоя, когда пороги квантования выбираются для каждого фильтра в отдельности. Такой подход сейчас является стандартом для квантования свёрточных слоёв нейронной сети из-за того, что он позволяет получать меньшую просадку по точности при незначительном увеличении времени обработки.

Квантование при обучении или квантование после обучения

Часто при квантовании параметров нейронной сети необходимо её дообучение, чтобы минимизировать просадку точности. Такое дообучение нейронной сети может быть выполнено одновременно с квантованием или не выполнено вовсе.

В первом случае нейронная сеть обучается исходя из знания, что её параметры в дальнейшем будут квантоваться. В таком случае производится эмуляция того, что параметры сети уже квантованные на этапе обучения. Для этого при прямом проходе к её параметрам применяется операция ложного квантования, квантующая значения, но не препятствующая протеканию через них градиентов для обучения такой сети. Так как градиент операции квантования равняется 0 почти везде, применяют технику с заменой его на линейную функцию [62] (Рис. 1.8).

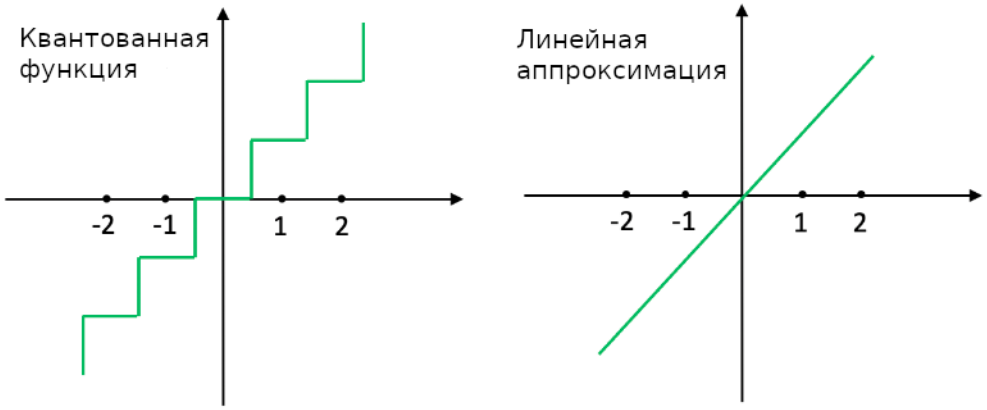

Рисунок 1.8 — Представление значений функции квантования и её линейной аппроксимации для вычисления градиента.

Несмотря на достаточно грубое линейное приближение функции квантования обучение нейронной сети из расчёта на её дальнейшее квантование - довольно эффективная техника, позволяющая минимизировать просадки по точности. Однако недостатком такого подхода является необходимость дообучения сети, что требует дополнительных вычислительных ресурсов. Как правило, нейронная сеть обучается до сходимости без квантования, затем в неё добавляются операции квантования и после этого она какое-то время дообучается с целью корректировки весов.

Альтернативным подходом к увеличению точности сети является выбор порогов квантования после обучения таким образом, чтобы уточнение параметров сети не требовалось [59]. Такой подход гораздо менее требователен к вычислительным ресурсам и времени на квантование, но позволяет получить меньшую точность итоговой модели, чем при квантовании с дообучением.

Квантование как инструмент реализован в большинстве популярных фреймворков для обучения нейронных сетей, таких как PyTorch [44] и TensorFlow [43]. Последний получил большее распространение для оптимизации сетей для мобильных устройств, ввиду того что его дочерний фреймворк TFLite более эффективно использует ресурсы вычислителя и позволяет выполнять нейронный сети с большей скоростью. Для квантования моделей Tensorflow использует подход асимметричного статичного послойного квантования, требующий дообучения модели [58]. Предложенный в [58] метод производит квантование в 8 бит и получил широкую популярность из-за эф-

Таблица 2 — Сравнение точности и времени работы сети MobileNet SSD до и после квантования

Тип данных	mAP	Малая (мс)	Большая (мс)
Float	22.1	778	370
8 bits	21.7	687	272

фективной его реализации на мобильных устройствах. Также данная работа исследует квантование мобильных архитектур нейронных сетей, в отличии от предшественников, выполнявших квантование для больших и излишне параметризованных архитектур. В частности, была продемонстрирована эффективность квантования для архитектуры MobileNet SSD [63; 64] для задачи детекции объектов на датасете COCO [6]. Даже для архитектуры, изначально оптимизированной для мобильного телефона, данный подход позволил получить минимальное снижение среднего AP (mAP, описывается в разделе 2.4), показав существенный выигрыш по скорости выполнения (Таблица 2).

Квантование как метод оптимизации производительности и потребляемой памяти сети применим для всех классов задач, решаемых нейронными сетями. И может быть эффективно применён на завершающих этапах подготовки нейронной сети для развёртывания на конечном устройстве. При этом при разработке сети можно не учитывать факт её дальнейшего квантования и применять его после других методов ускорения, таких как прунинг и дистилляция.

1.5 Нейросетевой архитектурный поиск

Задание архитектуры нейронной сети сложный процесс, включающий выбор количества слоёв, их типа и гиперпараметров каждого слоя. Для задания эффективной архитектуры требуется привлечение эксперта и большое количество экспериментов по выбору параметров. Но даже в таком случае получаемая нейронная сеть может иметь субоптимальную архитектуру.

Минимизировать человеческий фактор и оптимизировать процесс выбора архитектуры помогают методы нейросетевого архитектурного поиска [32—34] (НАП). Такие подходы позволяют в автоматическом режиме, с учётом домена работы сети и ограничений на её точность и скорость, выбрать оптимальную архитектуру. В данном случае под доменом работы понимаются данные, подаваемые на вход нейронной сети и значения, которые ей необходимо предсказать. От человека при этом требуется задать:

- пространство поиска;
- метод оптимизации;
- метод оценки архитектуры.

Пространство поиска определяет базовые операции архитектуры: слои и их возможные значения гиперпараметров. Метод оптимизации задаёт оптимизационную процедуру, согласно которой из пространства поиска будут выбираться слои формирующие архитектуру. При этом архитектура выбирается таким образом, чтобы найти экстремум значения метрики получаемой методом оценки архитектуры. Такой метод может, например, максимизировать значение точности модели или максимизировать скорость её работы или пытаться найти баланс между этими показателями.

Одной из ключевых работ по НАП является [32]. В данной работе предлагается рекуррентная модель, обученная методом обучения с подкреплением искать оптимальную архитектуру сети. Данная работа простимулировала интерес к методам НАП. Существующие на сегодняшний день подходы можно классифицировать следующим образом:

- По пространству поиска:
  - С фиксированным порядком слоёв;
  - Блочные;
  - Иерархичные;
  - На основе морфизма;

– По методу оптимизации:
    – Эволюционные алгоритмы;
    – Обучение с подкреплением;
    – Градиентная оптимизация;
    – Суррогатные модели.

Пространство поиска.

Классификация по пространству поиска осуществляется на основе того, каким образом задаётся пространство возможных слоёв для архитектуры нейронной сети. Архитектура нейронной сети задаётся как направленный ациклический граф, состоящий из слоёв. Сложность использования методов НАП связана именно с заданием подходящего пространства слоёв. Такое пространство поиска в наибольшей степени определяет эффективность архитектуры, которую можно получить алгоритмом НАП.

Наиболее простой способ задать пространство поиска заключается в задании архитектуры с фиксированным порядком слоёв: положение каждого узла и размер его входных и выходных тензоров заранее заданы вычислительным графом. При этом поиск осуществляется по пространству возможных слоёв, используемых на месте каждого вычислительного узла в графе. Таким образом пространство поиска задаётся списком возможных слоёв для каждого узла вычислительного графа. Однако такой подход имеет существенный недостаток - плохая масштабируемость получаемой архитектуры. В большинстве прикладных задач от объёма доступного набора данных зависит число необходимых обучаемых параметров в нейронной сети. Поэтому объём архитектуры необходимо варьировать в зависимости от числа доступных данных с сохранением основных архитектурных особенностей.

Решением данной проблемы служит блочный тип задания архитектуры нейронной сети. В таком случае архитектура задаётся как некоторое количество повторяющихся архитектурных блоков. Следовательно, задача НАП сводится к поиску архитектуры такого блока, что упрощает пространство поиска. При этом, варьируя число блоков в модели, можно регулировать глубину сети и число обучаемых параметров, подстраиваясь под различные наборы данных. Так в [33] демонстрируется, что модель, найденная с помощью НАП на набо-

ре данных CIFAR-10, демонстрирует точность сравнимую с точностью моделей, разработанных экспертами на наборе данных ImageNet. Большинство современных исследований связанных с НАП на основе блочного пространства поиска основывается на работе [34], авторы которой одними из первых предложили такой подход. Однако такой подход связан со сложностями при практическом применении. Так при поиске архитектур в методе DARTS [33] используется сеть малой глубины, что вызвано ограничением памяти GPU. При этом, при большой разнице в числе блоков во время поиска и при исполнении архитектуры, эффективность найденных блоков может снижаться. Для снятия данного ограничения в [65] был предложен P-DARTS постепенно наращивающий число блоков в модели во время процедуры поиска.

Развитием идеи представления архитектуры в виде набора повторяющихся блоков является иерархическое пространство поиска. В данном случае поиск осуществляется на двух уровнях: поиск архитектуры блока и поиск архитектуры сети, состоящей из найденных блоков. При этом на втором этапе в качестве элементарного вычислительного узла рассматриваются найденные блоки (Рис. 1.9).

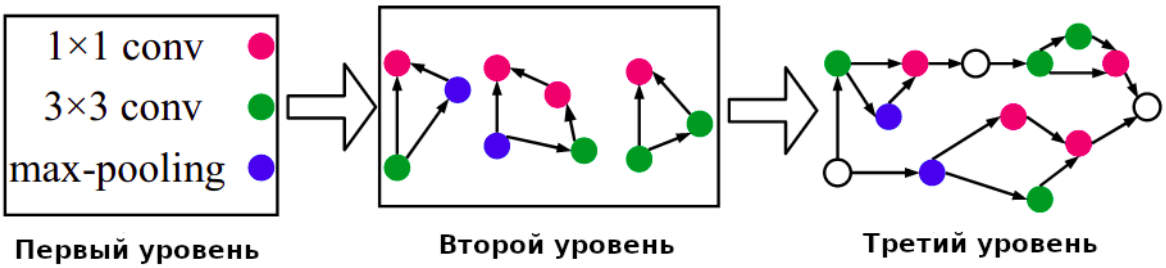

Рисунок 1.9 — Схематическое изображение НАП с иерархическим пространством поиска

Подход, в котором высокоуровневые блоки архитектуры строились итеративно на основе более низкоуровневых блоков, был предложен в [66] и в [67]. Однако такой подход обладает следующим недостатком: после того, как архитектура блока задана, он используется повсеместно во всей нейронной сети без возможности его изменить. Одно из решений данной проблемы было предложено в MnasNet в [68] и заключалось в использовании факторизованного пространства поиска для получения различных блоков для различных

участков сети. Другим значительным ограничением являлось использование прокси набора данных для поиска архитектуры блока. В [69] предлагается подход ProxylessNas для поиска архитектуры блоков и общей архитектуры сети на общем наборе данных.

Альтернативным подходом к заданию пространства поиска являются подходы на основе морфизма. Данная техника [70; 71] в некотором роде похожа на процедуру дистилляции и прунинга, так как позволяет оптимизировать производительность уже обученной сети. Подход заключается в добавлении в обученную сеть операций морфизма, которые позволяют регулировать ширину и глубину сети.

Методы оптимизации.

Описанные выше методы позволяют определить пространство возможных слоёв, по которым осуществляется поиск архитектуры, заключающийся в минимизации целевой метрики. Саму процедуру поиска можно осуществлять с помощью нескольких методов оптимизации.

Наиболее ранний из таких методов — это эволюционные алгоритмы. Такие алгоритмы вдохновлены биологической эволюцией. Оптимизация осуществляется итерационно, на каждом шаге задаётся популяция различных архитектур и к следующему шагу переходят и скрещиваются наиболее "адаптированные". Распространённой практикой в таких подходах является задание архитектуры сети в виде "генома строки состоящей из 0 и 1, где 1 означает наличие связи между слоями, а 0 отсутствие. Сама же процедура эволюционного отбора осуществляется следующим образом:

1. Из популяции выбираются архитектуры с наилучшими показателями целевой метрики;
2. Происходит скрещивание архитектур: объединяются половина генома одной и другой архитектур;
3. В геном добавляются мутации: например, элементы генома случайно и независимо инвертируются (добавляются или удаляются связи между слоями).

Однако такой подход зависит от случайности и может потребовать большого числа итераций обновления популяции для выбора подходящей архитектуры.

К тому же требуется обучение большого числа моделей в каждой популяции, что накладывает серьёзные ограничения на необходимые для оптимизации ресурсы.

Альтернативой эволюционным алгоритмам служит обучение с подкреплением [32]. При таком подходе система поиска архитектуры состоит из контроллера и среды (Рис. 1.10).

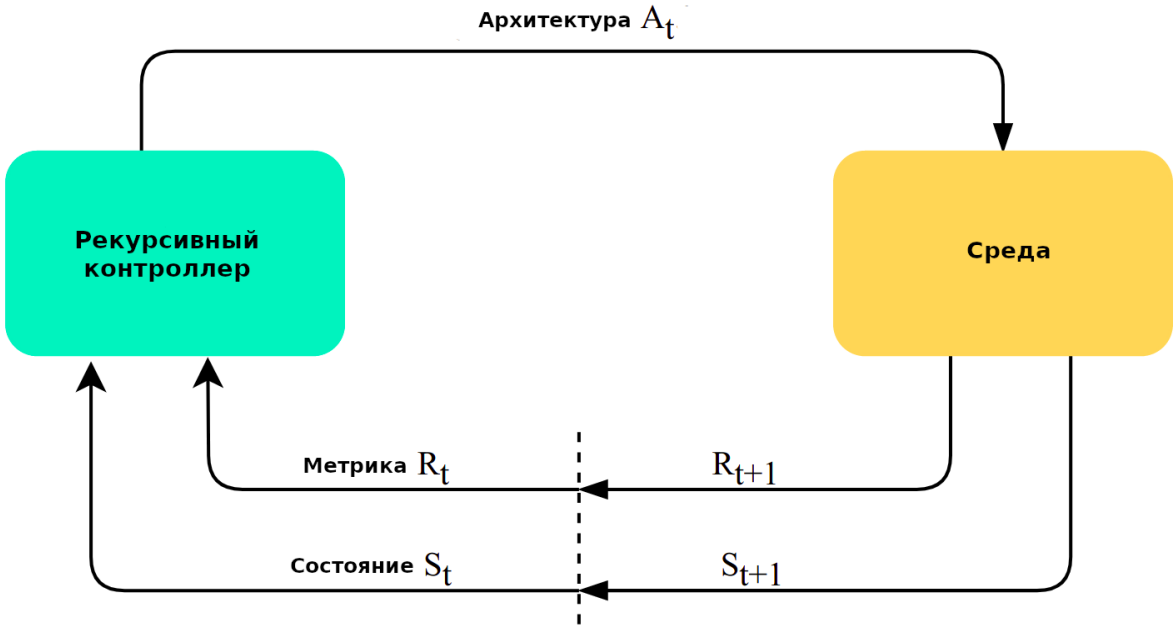

Рисунок 1.10 — Принцип работы системы выбора оптимальной архитектуры сети на основе обучения с подкреплением.

Среда представляет собой стандартный механизм обучения нейронной сети и измерение целевых метрик. В то время как контроллер, на основе полученных от среды значений метрик, пытается предсказать архитектуру их максимизирующую. Часто в качестве контроллера используется рекуррентная нейронная сеть, учитывающая предыдущие свои предсказания. Поиск оптимальной архитектуры на основе обучения с подкреплением как правило достаточно требователен к вычислительным ресурсам. Для решения данной проблемы в [72] предлагается использование асинхронной реализации и ранней остановки процедуры обучения нейронной сети.

Все описанные ранее методы оптимизации производили поиск архитектуры в дискретном пространстве, что являлось серьёзным ограничением. Снять

данное ограничение позволил предложенный в [33] метод DARTS, позволяющий осуществлять поиск архитектуры в непрерывном и дифференцируемом пространстве поиска. Это достигалось за счёт добавления в каждый узел архитектуры сети всех возможных в этом узле слоёв с дополнительным вектором обучаемых параметров от 0 до 1, каждый элемент которого обозначает вес прохождения через данный слой. Выход каждой операции в таком узле домножается на соответствующий вес, чтобы определить вклад операции в результат. Домножение происходит по следующей формуле:

$$\bar{o}(x) = \sum_{k=1}^{K} \frac{exp(\boldsymbol{\alpha}^k)}{\sum_{i=1}^{K} exp(\boldsymbol{\alpha}^i)} * o^k(x),$$

где значение выхода $\bar{o}$ узла для входа x вычисляется как взвешенная сумма всех возможных в этом узле типов слоёв $o^k(x)$ с нормированными весами $\boldsymbol{\alpha}^k$. Задание сети таким образом позволяет обучать её методами градиентного спуска. После обучения составленной таким образом сети, выбор финальной архитектуры осуществляется как выбор слоёв с наибольшим весом для каждого узла. При этом потребление требуемой памяти в таком подходе растёт линейно с увеличением числа возможных слоёв в каждом узле. Решение такой проблемы описывается в [69], где предлагается использовать бинарные вентили, позволяющие занимать память вычисляемыми градиентами только для активированного слоя.

Недавние работы предлагают решение ещё одной остро стоявшей для НАП проблемы - большое время поиска архитектуры. Такие подходы предлагают, так называемый, суррогатный метод оптимизации модели. При таком подходе на ранних этапах поиска архитектуры обучается дополнительная модель, предсказывающая возможную эффективность той или иной архитектуры. Данная модель затем используется при дальнейшем поиске архитектуры для отбора наиболее перспективных конфигураций без необходимости их обучать. Так в методе PNAS [67] используется LSTM для предсказания эффективности архитектуры переменного размера.

1.6 Выводы по первой главе

Все описанные в данной главе подходы комбинируются в общий алгоритм оптимизации нейронной сети. После постановки задачи процесс получения оптимальной для её решения архитектуры можно описать следующим образом:

1. Автоматический нейросетевой архитектурный поиск на основе целевого набора данных и оцениваемых метрик точности;
2. Дистилляция из обученной модели с большим числом параметров для передачи информации о сложноизвлекаемых признаках;
3. Прунинг полученной сети с целью удаления параметров, вносящих незначительный вклад в итоговую точность модели;
4. Квантование для понижения дискретизации весов и, как следствие, повышения вычислительной эффективности на конечном устройстве пользователя.

Однако, описанный выше алгоритм имеет нюанс, который необходимо учитывать при его применении. Если на первом шаге автоматическим нейросетевым архитектурным поиском была найдена достаточно оптимальная архитектура, производить третий шаг, как правило, не целесообразно. Как было оговорено ранее, алгоритмы прунинга эффективно работают только с избыточно параметризованными моделями, в то время как найденная архитектурным поиском нейронная сеть уже содержит оптимальное количество параметров. Так как задача распознавания лиц средствами нейронных сетей актуальна и, как следствие, достаточно широко распространена – для таких нейронных сетей разработаны специализированные архитектуры, полученные с применением архитектурного поиска и глубокого экспертного анализа [21;22]. Таким образом, применение прунинга кажется неэффективным и не будет рассматриваться в данной диссертации.

Вместо этого в последующих главах будет подробно рассмотрена задача дистилляции, так как она позволяет увеличивать точность сетей с малым числом параметров, как было описано в данной главе ранее. Однако задачи биометрии, и распознавания лиц в частности, требуют обучения нейронных

сетей, производящих разнесение векторов в векторном пространстве и использующих для этого особые функции ошибки. Для таких функций ошибки задача дистилляции не решена в полной мере и подробно изучается в данной диссертации. Также в данной работе предлагается новый специализированный метод дистилляции для задачи распознавания лиц, позволяющий получать большую точность, чем существующие подходы.

Хотя описанный выше общий алгоритм оптимизации производительности нейронных сетей позволяет получать оптимальную архитектуру, в некоторых задачах могут возникать издержки по времени, не связанные со временем работы сети. Примером таких издержек в задаче распознавания лиц могут стать ложные срабатывания детектора лиц, так как каждое срабатывание детектора влечёт за собой построение биометрического вектора и сравнение его с базой лиц. А это, в свою очередь, увеличивает нагрузку на вычислитель и время отклика, а при хранении базы лиц на сервере может приводить к существенному удорожанию системы из-за вынужденного увеличения пропускного канала сетевого соединения с сервером и мощности сервера для обработки всех поступающих запросов. Для решения проблемы ложных срабатываний детектора в данной диссертации предлагается метод, рассматриваемый в следующей главе.

Последний пункт алгоритма оптимизации нейросетевой модели – квантование, может быть применено ко всем описанным в данной диссертации нейронным сетям и не требует разработки особых методов ввиду общности существующих подходов. Поэтому квантование не будет подробно разбираться в настоящей диссертации.

Таким образом, основной целью диссертации является исследование задачи дистилляции и снижения числа ложных срабатываний детектора людей, как основных узких места в задаче построения быстрого и точного механизма распознавания лиц на встраиваемых устройствах.

Глава 2. Оптимизация детектора лиц

2.1 Задача детектирования лиц

Задача детектирования лиц на изображении заключается в восстановлении координат ограничивающего лицо прямоугольника и уверенности сети в наличии в данной области лица (предсказанной вероятности обнаружить лицо в данной области). Как правило, детекторы возвращают некоторое фиксированное число таких пар (ограничивающая рамка и вероятность) для каждого изображения.

В данной главе предлагается разработанный в рамках диссертационного исследования метод детектирования лиц, позволяющий снизить число ложных срабатываний за счёт обработки только движущихся объектов. Предлагаемый метод заключается в модификации уже обученной на детекцию СНС и может быть применён для повышения точности некоторой имеющейся системы путём небольших изменений в ней.

К настоящему моменту детекторы на основе нейронных сетей зарекомендовали себя как наиболее точный инструмент для обнаружения объектов в видеопотоке или на отдельном кадре [73—75]. В последнее время нейросетевые детекторы достигли достаточно высоких скоростей исполнения в маломощных системах [76—78]. Так нейросетевые детекторы позволяют организовать отслеживание дорожного трафика или систему наружного наблюдения на основе одноплатного компьютера. Снижение нагрузки на такой вычислитель позволит продлить срок его эксплуатации. В задачах, связанных с детектированием, будь то дорожный трафик или люди для системы наружного наблюдения, кажется нецелесообразным обрабатывать статичные кадры, так как состояние окружения не поменялось. В диссертации предлагается подход, позволяющий эффективно и с высокой точностью отфильтровывать статичные кадры из видеопотока, без дополнительной вычислительной нагрузки.

Также для систем распознавания лиц ложные положительные срабатывания детектора могут оказывать значительное негативное влияние на работу всей системы. Ложные срабатывания детектора могут приводить к недетерминированному поведению, когда система пытается обработать фрагмент фона как лицо. Часто в таких случаях применяют детектор движения и обрабатывают только те кадры, на которых есть движущиеся объекты [79; 80]. Как правило, такие подходы для статичной камеры основываются на построении модели фона и вычислении разницы с ней. В предлагаемом подходе этапы построения устойчивой модели фона и детектирования объединяются. Для этого используются промежуточные карты признаков детектора как модель фона. Такой подход не требует дополнительных вычислений и позволяет получить устойчивую к шуму модель фона.

Для решаемой задачи отсеивания статичных кадров и для отфильтровывания ложных срабатываний детектора не надо отвечать на вопрос сегментации движения с точностью до пикселя [81; 82], достаточно понимать есть ли движение внутри области детекции. Это позволяет оптимизировать процедуру обнаружения движения рассмотрением решётки заданного масштаба, соответствующей карте признаков промежуточного слоя СНС.

Идея использования промежуточной карты признаков для сравнения кадров в видеопотоке близка к идее perceptual loss [83], широко используемой для обучения генеративных нейронных сетей [84; 85]. В таких подходах для определения схожести, генерируемого и целевого изображений, используются карты признаков от некоторой свёрточной сети, такой как VGG-19 [14]. В данной диссертации предлагается использовать промежуточные карты признаков той же сети детектора, что не увеличивает вычислительную нагрузку.

Биологическим прототипом данного подхода послужила система зрения земноводных. При отсутствии изменения в наблюдаемом мире, количество собираемой земноводным зрительной информации куда меньше, чем у млекопитающих. Амфибия замечает только движущиеся объекты. Аналогично предлагаемая в данной диссертации модель машинного зрения реагирует только на движущиеся объекты, что позволяет снизить вычислительную нагрузку.

2.2 Методы детектирования лиц

Детектирование объектов - одна из основных задач машинного зрения. Наибольшее распространение получили модели на основе R-CNN [73—75], YOLO [76—78], SSD [64].

Faster R-CNN. Модели для детектирования на основе R-CNN использует генератор регионов, предположительно содержащих объект, и СНС для их классификации. Классификация применяется для разделения регионов на содержащие и не содержащие интересующие категории объектов. S. Ren и соавторы [75] предложили подход Faster R-CNN, использующую Region Proposal Network. Данная сеть использует те же карты признаков что и детектор, что значительно снижает вычислительные затраты на генерацию регионов.

You only look once. J. Redmon и соавторы [76] были первыми, кто предложили подход к выполнению детектирования одной нейронной сетью. YOLO разделяет входное изображение на решётку, где для каждой ячейки предсказываются ограничивающие объект рамки и вероятности обнаружения классов. Для предсказания координат рамки и вероятностей классов используются полносвязные слои. Такой подход не содержит этапа генерации гипотез и инкапсулирует всю логику внутри одной нейронной сети. Это позволило получить высокую скорость работы данной архитектуры. В YOLOv2 [77] авторы заменили полносвязный слой на использование якорных ограничивающих рамок (anchor boxes). В YOLOv3 [78] авторы использовали более точную архитектуру нейронной сети для извлечения признаков и улучшили детектирование объектов разного размера.

SSD: Single Shot MultiBox Detector. Wei Liu и соавторы [64] предложили другой подход к детектированию объектов на изображении, используя одну нейронную сеть. SSD генерирует шаблонные ограничивающие рамки с различным соотношением сторон и различными масштабами для карты признаков изображения. Затем нейронная сеть предсказывает вероятности обнаружения целевых классов в шаблонных ограничивающих рамках и уточняет их координаты. Основным отличием от YOLO является использование карт признаков

с различных слоёв, что повышает точность детектирования объектов различного размера.

Для снижения числа ложных срабатываний детектора система может использовать априорное знание о том, что подходящий к домофону человек - это движущийся объект. Задача отделения движущихся объектов от фона давно вызывает интерес у исследователей, занимающихся машинным зрением. Общий подход к обнаружению движения на изображении можно описать двумя основными шагами. На первом шаге производится построение модели фона - статичной части изображения. На втором шаге обнаруживается движение как разница между текущим кадром и моделью фона. Классические подходы к обнаружению движения использовали в качестве модели фона опорный кадр. Широкое распространение получила Gaussian Mixture Model [79] предложенная Stauffer C. И Grimson W. E. L. Такой подход учитывал изменения в модели фона, связанные с динамическим фоном и изменением условий освещения. St-Charles P. L. и соавторы [80] предложили подход SuBSENSE, который использует пространственно-временные бинарные признаки и цветовую информацию для обнаружения изменений. Wang Y. и соавторы [86] предложили открытый набор тестовых данных для обнаружения движения с точностью до пикселя в различных условиях.

Повысить точность обнаружения движения в видеопотоке позволили методы глубокого машинного обучения. Считается, что Braham M. и соавторы [87] первыми предложили использовать СНС для построения модели фона. Для этого они строили модель фона в градациях серого на основе N кадров изображения. Затем обучали зависящую от рассматриваемой сцены СНС модель по модели фона и текущему кадру сегментировать движущиеся объекты. Babaee M. и соавторы [81] развили идею использования СНС для детектирования движения на изображении. Они использовали SuBSENSE [80] и Flux Tensor [88] для построения модели фона, затем, подавая на вход СНС модель фона и текущий кадр, с точностью до пикселя обнаруживали движение. Ang Lim L. и Yalim Keles H. [89] предложили FgSegNet, в основе которого лежит СНС на основе триплетов, использующая различные масштабы изображения для кодирования. Использование различных масштабов позволило им учитывать контекст, в ко-

тором находится фрагмент на одном масштабе, с помощью другого масштаба. Затем они с помощью декодировщика обнаруживали движение с точностью до пикселя. В своей последующей работе Ang Lim L. и Yalim Keles H. [82] предложили FgSegNet-v2. Данный метод расширял модуль объединения признаков FgSegNet введением функций признаков внутрь модели, что позволило извлекать признаки различного масштаба из изображения.

Недостатком существующих подходов на основе СНС является необходимость обучать дополнительную нейронную сеть, что увеличивает ресурсы, необходимые для подготовки и развёртывания такой системы. Использование дополнительной СНС может негативно сказываться на времени обработки кадра. В данной диссертационной работе предлагается подход, не требующий дополнительных вычислительных ресурсов для обнаружения движения, так как для сравнения кадров используются карты признаков из промежуточного слоя сети детектора.

Объединение детектора объектов и движения. Для повышения скорости работы детектора его могут объединять с детектором движения. При таком подходе кадры, не содержащие движущихся объектов, не будут обрабатываться детектором. Kang D. и соавторы предложили подход Noscope [90]. Он использует архитектурный поиск модели, оптимальной для каскада детекторов, обнаруживающих разницу в кадрах, и специализированных сетей для заданного видео, целевого объекта и опорной нейронной сети. Использование в качестве детектора разницы кадров среднеквадратичной ошибки (MSE) между опорным кадром и текущим позволило получить высокую скорость обработки видеопотока. Точность полученного каскада при этом совпадает с точностью опорной нейронной сети. Однако такой подход имеет несколько недостатков. Во-первых, он требует переобучения системы для каждой камеры и сцены, в отличие от рассматриваемых ранее методов детекции, таких, как SSD, YOLO и R-CNN. Во-вторых, обнаружение движения на основе сравнения с опорным кадром неустойчиво к динамическому фону и изменению освещения.

Другой подход к совмещению обнаружения движения и детектора объектов предложили Yu R., Wang H. и Davis L. S в методе ReMotENet [91]. Они задались задачей обнаруживать движение объекта определённого класса на ви-

део. Для этого они анализируют видео целиком с помощью трёхмерных (3D) свёрток. Недостатком такого подхода является необходимость обрабатывать видео целиком, что ограничивает его применимость для камер наружного наблюдения. Также ReMotENet отвечает только на вопрос, было ли движение объекта интересующего класса на видео, но не локализует его.

Предлагаемый здесь подход использует карту признаков, полученную от промежуточных слоёв нейронной сети детектора. Это позволяет производить сравнение кадров на основе сложных иерархий признаков, найденных СНС. В случае отсутствия движения обработка кадра сетью останавливается на этом этапе.

2.3 Разработка устойчивого подхода к обнаружению движения

В данной диссертационной работе предлагается подход к детектированию движущихся объектов в видеопотоке (Рис. 2.1) - AmphibianDetector. Входное изображение обрабатывается первыми m слоями извлекателя признаков, входящим в состав модели детектирования объектов. Полученные на этом этапе карты признаков используются для построения модели фона. Карта признаков для текущего кадра сравнивается с моделью фона по косинусной близости. Полученная после этого карта движений используется для проверки необходимости заканчивать выполнение детектора. Карта движений также используется для отфильтровывания ограничивающих рамок, чтобы отбраковать ложные срабатывания, соответствующие статичным объектам. Этот метод позволяет сократить среднее время обработки кадра видеопотока и снизить число ложных срабатываний. Ключевая идея предлагаемого метода - использование промежуточных карт признаков из нейронной сети детектора для определения регионов движения в кадре. При этом карта признаков промежуточного слоя рассматривается как набор векторов, где каждый вектор описывает соответствующую область на входном изображении. Затем, сравнивая эти вектора для текущего

кадра и вектора, представляющие модель фона, можно оценить изменения в различных участках изображения.

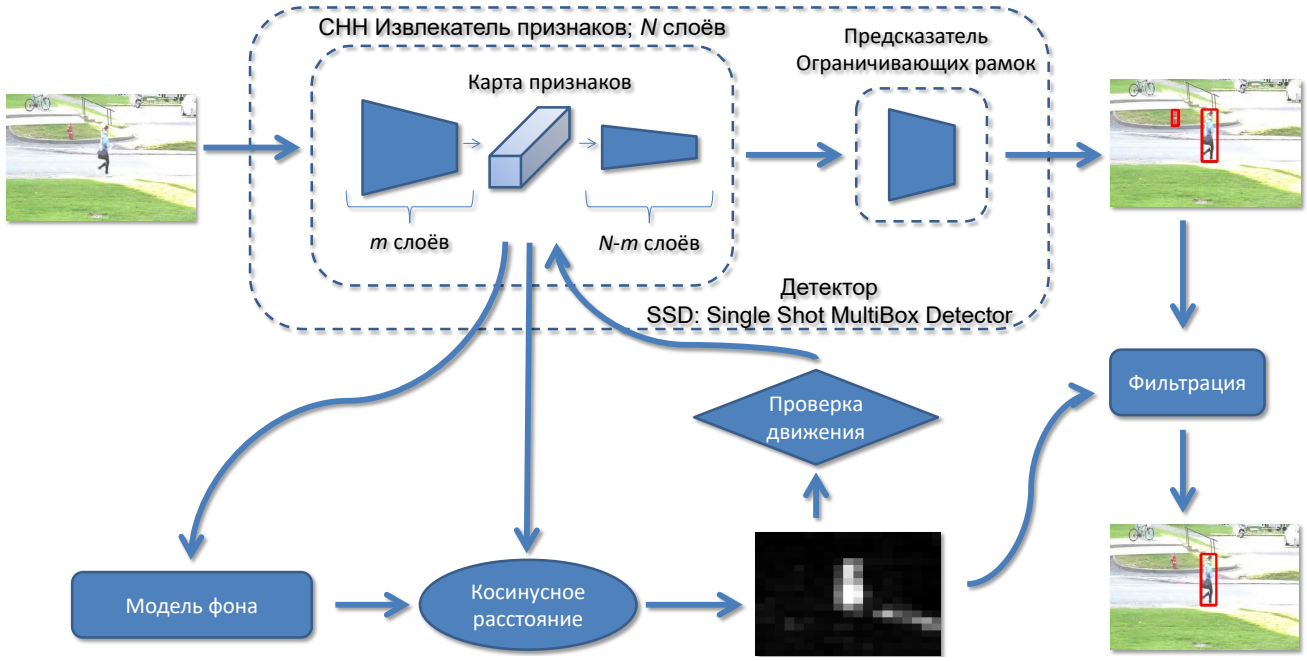

Рисунок 2.1 — Схема устойчивого к зашумлённости входных данных метода для обнаружения движения в видеопотоке на основе промежуточных карт признаков СНС.

Обозначим через $F_1..F_i..$ последовательность RGB кадров в видеопотоке, где F_i соответствует i-ому кадру. В качестве модели детектирования объектов возьмём SSD [64], как наиболее распространённую архитектуру нейронной сети для обнаружения объектов. Архитектура сети для детектирования объектов состоит из извлекателя признаков (feature extractor) из изображения и части сети предсказывающей координаты ограничивающих рамок. Извлекатель признаков представляет собой свёрточную нейронную сеть такую, как MobileNetV2 [92] или Inception [93]. Она задаётся N свёрточными слоями $L_1..L_N$. Для получения карты признаков с промежуточного слоя выбирается некоторое $m \in [1..N]$. В качестве карты признаков предлагается использовать выход L_m слоя нейронной сети. Карту признаков для кадра i будем обозначать через M_i. Получаемая карта признаков имеет размер $H_m \times W_m \times C_m$, где H_m и W_m соответствуют высоте и ширине карты признаков. C_m задаёт число каналов и может интерпретироваться как размерность вектора в каждой точке карты признаков.

От выбора m зависит какой области входного изображения будет соответствовать каждый вектор в карте признаков, а также то, насколько устойчиво векторное представление к артефактам входного изображения. Регулируя параметр m, можно добиваться нужного соотношения между скоростью и точностью работы детектора движений. Выбор оптимального m будет подробно рассмотрен в следующем разделе.

Размер участка входного изображения, соответствующего одному вектору на карте признаков для заданного m, вычисляется следующим образом. Пусть $H_{in} \times W_{in} \times 3$ размер трёхканального входного изображения, где H_{in} и W_{in} соответственно высота и ширина входного изображения. Тогда, для заданного m, размер входной области будет задаваться как $H_{in}/H_m \times W_{in}/W_m$.

Для обнаружения изменений в F_i кадре построенная для него карта признаков M_i сравнивается с моделью фона M_{bg}. Модель фона инициализируется некоторым опорным кадром M_{init}. Опорный кадр выбирается таким образом, чтобы он не содержал движущихся объектов, так как их смещение будет вызывать ложные срабатывания.

Для сравнения модели фона M_{bg} и карты признаков кадра M_i вычисляется косинусная близость между представляющими их векторами. Для точки пространства x, y, косинусная близость вычисляется как:

$$Diff_i = 1 - \frac{M_{bg}[x,y] \times M_i[x,y]}{||M_{bg}[x,y]|| * ||M_i[x,y]||}$$

где $\times$ - скалярное произведение векторов, а $*$ - произведение норм векторов. Значения карты движения $Diff_i$ тем ближе к 1, чем сильнее изменение на участке входного изображения F_i. Как отмечают Wang H. и соавторы [52], использование косинусной близости для сравнения векторных представлений, обученных с использованием функции softmax, повышает согласованность в сравнении с Евклидовым расстоянием.

Оптимизация скорости работы детектора производится за счёт остановки обработки кадров, на которых не обнаружено движение. Для отбраковки статичных кадров максимальное значение $max(Diff_i)$ сравнивается с порогом λ. Кадры F_i, для которых $max(Diff_i)$ меньше порога λ, используются для обновления модели фона.

Как отметили Babaee M. и соавторы [81], динамическое обновление модели фона позволяет адаптироваться к изменениям условий освещённости сцены и погодных условий. В рассматриваемом подходе M_{bg} обновляется на основе M_i по следующей формуле:

$$M_{bg} = \begin{cases} M_{bg} * \alpha + M_i * (1 - \alpha) & \text{если } max(Diff_i) < \lambda \\ M_{bg} & \text{иначе} \end{cases}$$

Регулируя параметр α, можно настраивать то, насколько долго объекту нужно неподвижно оставаться в кадре, чтобы стать частью модели фона. Для $\alpha = 0$ каждый кадр будет сравниваться с картой признаков предыдущего кадра. Случай для $\alpha = 1$ соответствует сравнению карты признаков i-го кадра с опорным.

Для кадров с движущимися объектами после предсказания ограничивающих рамок производится дополнительная фильтрация на основе $Diff_i$. Координаты ограничивающих рамок предсказываются в нормированных значениях от 0..1 и затем масштабируются на размер входного изображения $F_i : H_{in} \times W_{in}$ и карту движения $Diff_i : H_m \times W_m$. Обозначим ограничивающие рамки для кадра F_i через $B_{i1}..B_{iM}$. Тогда для ограничивающей рамки с индексом k координаты, соответствующие области на входном изображении, будут задаваться как:

$$Bin_{ik} = [B_{ik}.left * W_{in}, B_{ik}.top * H_{in}, B_{ik}.right * W_{in}, B_{ik}.bottom * H_{in}].$$

Для карты движений $Diff_i$ соответственно:

$$Bdiff_{ik} = [B_{ik}.left * W_m, B_{ik}.top * H_m, B_{ik}.right * W_m, B_{ik}.bottom * H_m].$$

Bin_{ik} удаляется из списка ограничивающих рамок если $mean(Bdiff_{ik})$ меньше порога λ. Здесь среднее вычисляется по $Diff_i$ при варьировании x, y в области, заданной $Bdiff_{ik}$.

Предложенный метод позволяет получить ускорение обработки потокового видео за счёт остановки обработки статичных кадров, а также снизить число ложных срабатываний детектора для задач, в которых необходимо детектировать движущиеся объекты.

2.4 Используемые метрики

Метрика **Mean Average Precision (mAP)** широко используется для оценки точности работы детекторов объектов. Для вычисления значения этой метрики вводится критерий обнаружения детектором заданного объекта как превышение значения IoU заданного порога. Intersection over Union (IoU) - измеряет точность соответствия предсказанного ограничивающего прямоугольника и прямоугольника заданного в разметке набора данных. IoU вычисляется по следующей формуле:

$$IoU(A, B) = \frac{S_{intersection}(A, B)}{S_{union}(A, B)},$$

для ограничивающих прямоугольников A и B, где $S_{intersection}$ и S_{union} означают площади областей пересечения и объединения прямоугольников, как показано на рисунке 2.2.

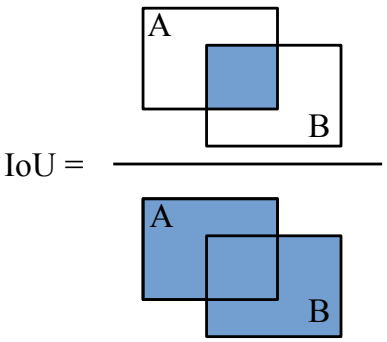

Рисунок 2.2 — Вычисление значения IoU как отношения пересечения к объединению.

Если значение IoU с разметкой набора данных больше заданного порога (как правило 0.5) объект считается успешно обнаруженным. Имея критерий детектирования объекта, мы можем применять метрики бинарной классификации, такие как точность (precision) и полнота (recall):

$$Precision = \frac{TP}{TP + FP};$$

$$Recall = \frac{TP}{TP + FN},$$

где TP - верно задетектированные (true positive), FP - ложноположительные срабатывания детектора (false positive), FN - ложнонегативные срабатывания детектора (false negative).

Значение Average Precision (AP) вычисляется как площадь под Precison-Recall кривой:

$$AP = \int_0^1 \text{Precision}(\text{Recall})\, d\,\text{Recall},$$

иными словами как среднее значение точности при варьировании полноты выбором порога уверенности детектора для предсказанных вероятностей. Mean Average Precision (mAP), в свою очередь, вычисляется как среднее значение AP для всех классов, обнаруживаемых детектором:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i.$$

mAP учитывает баланс между полнотой и точностью детектора и рассматривает как ложноположительные, так и ложнонегативные срабатывания. Это делает mAP подходящей метрикой для оценки точности работы детекторов.

2.5 Экспериментальная апробация предложенного подхода

Было проведено сравнение предложенного подхода AmphibianDetecor с базовым подходом к решению задачи отфильтровывания ложных срабатываний на статичные объекты.

Базовый подход решения заключается в попиксельном сравнении кадров по L_2 расстоянию для определения движения. Для вычисления модели фона в базовом подходе использовалось задание опорного кадра и обновление модели с коэффициентом α, как описано в предыдущем разделе. В экспериментах оценивались скорость и точность предложенного алгоритма. Для оценки скорости вычислялось среднее время обработки кадра для всех видео в рассматриваемом наборе данных. Точность оценивалась как метрика mean Average Precision (mAP) [4]. В качестве набора данных для оценки алгоритма рассматривался

поднабор данных с пешеходами - pedestrian из CDNet2014 [86]. Выбор в пользу этого набора данных был сделан по причине его открытости и близости задачи к детектированию лиц в видеопотоке. Данный поднабор данных содержит 10 видео с движением пешеходов при различном освещении и в различных погодных условиях. Также набор данных содержит видео, снятые внутри и снаружи помещений. Всего 26248 кадров, в среднем 2624 кадров на видео.

Рисунок 2.3 — a) Пример исходного кадра из видео; b) Кадр из видео после добавления человекоподобного объекта и шума.

Была проведена симуляция случаев, когда детектор осуществляет ложное срабатывание на статичный объект, схожий с объектом целевого класса. Для этого в кадры исходного видео было добавлено статичное изображение объекта, напоминающее человека (Рис. 2.3). Также видео, получаемые с камер наружного наблюдения, подвержены зашумлению из-за качества матрицы захватывающего устройства и канала передачи видео на сервер. Для того, чтобы приблизить набор данных к реальному случаю, каждый кадр модифицировался зашумлением. Исходные кадры умножались на Гауссовский шум с $\mu = 0.8$ и $\sigma = 0.2$, симулируя несовершенства устройства захвата видео.

Для сравнения AmphibianDetector с базовым подходом, эксперименты проводились на архитектуре SSD [64] с извлекателем признаков MobileNetV2 [92] со входом 300×300. Выбор данной архитектуры был обусловлен её популярностью для встраиваемых систем. Значения номера слоя m и порога λ выбирались поиском по сетке значений на одном видео из CDNet2014 pedestrian, состоящем

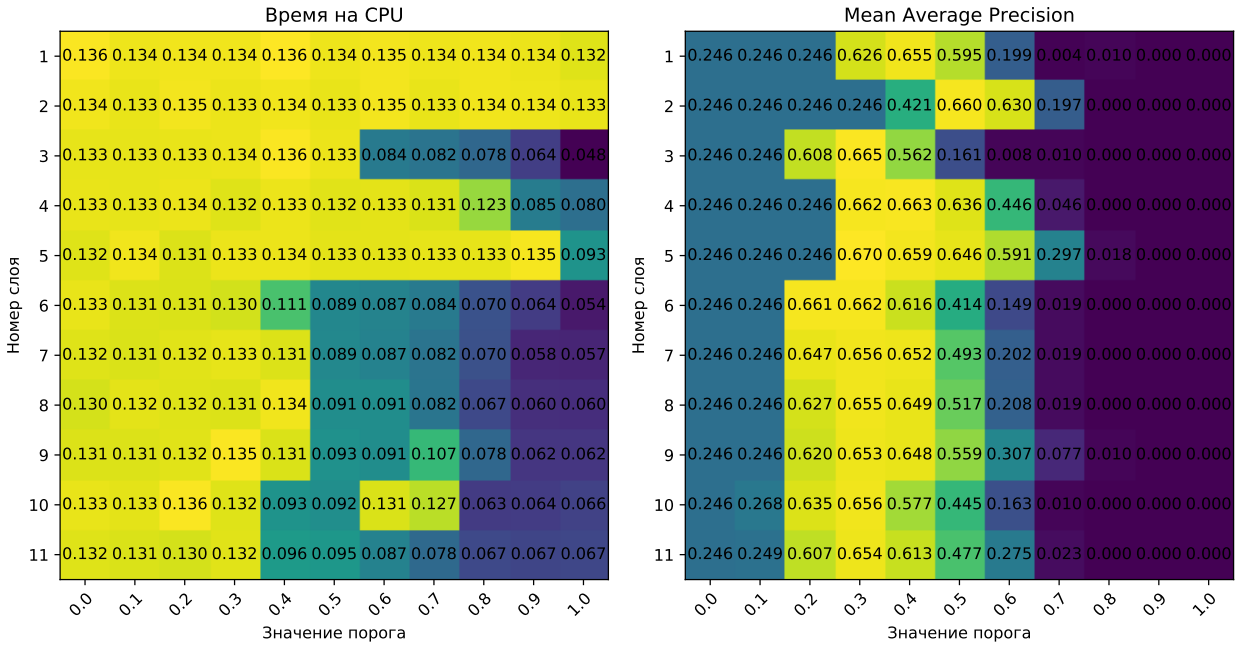

Рисунок 2.4 — Сравнение mAP и среднего времени обработки кадра на Intel Core i5-4210U CPU 1.70GHz × 4 для различных значений номера слоя m и значения порога λ.

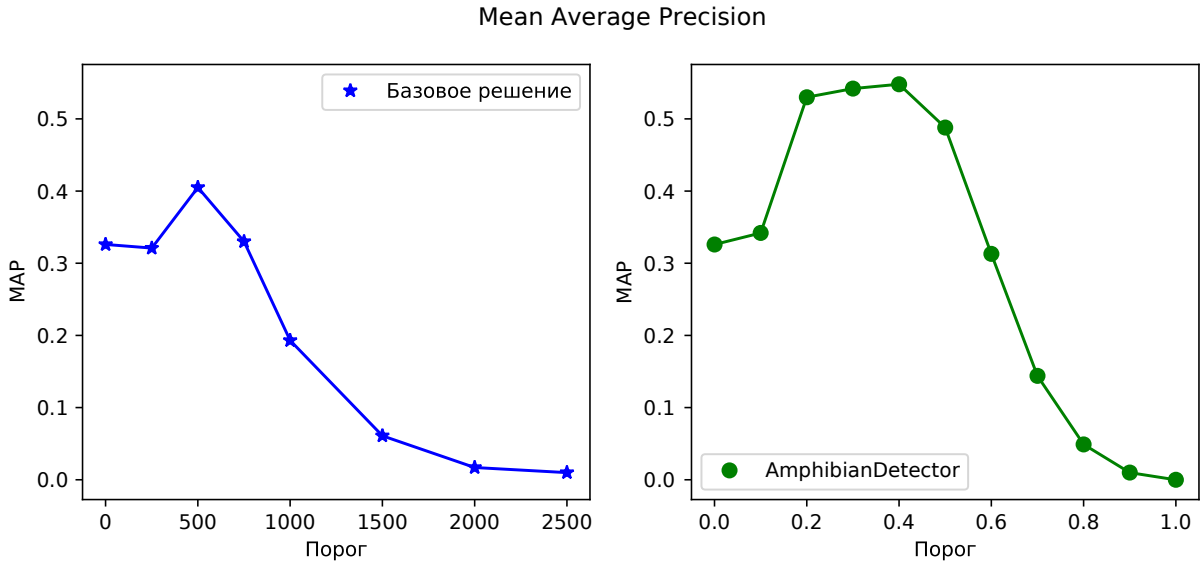

Рисунок 2.5 — Сравнение метрики mean average precision для AmphibianDetector и базового подхода на базе SSD+MobileNetV2.

из 2000 кадров. Порог λ варьировался от 0 до 1, так как это минимальное и максимальное значения для косинусной близости. Номер слоя m варьировался между 1 и 11 слоями извлекателя признаков. Из полученных тепловых карт (Рис 2.4) видно, что наибольшее значение mAP равное 0.67 достигается при

$m = 5$ и $\lambda = 0.3$. А наилучшие соотношение mAP к среднему времени обработки кадра при параметрах $m = 11$ и $\lambda = 0.4$.

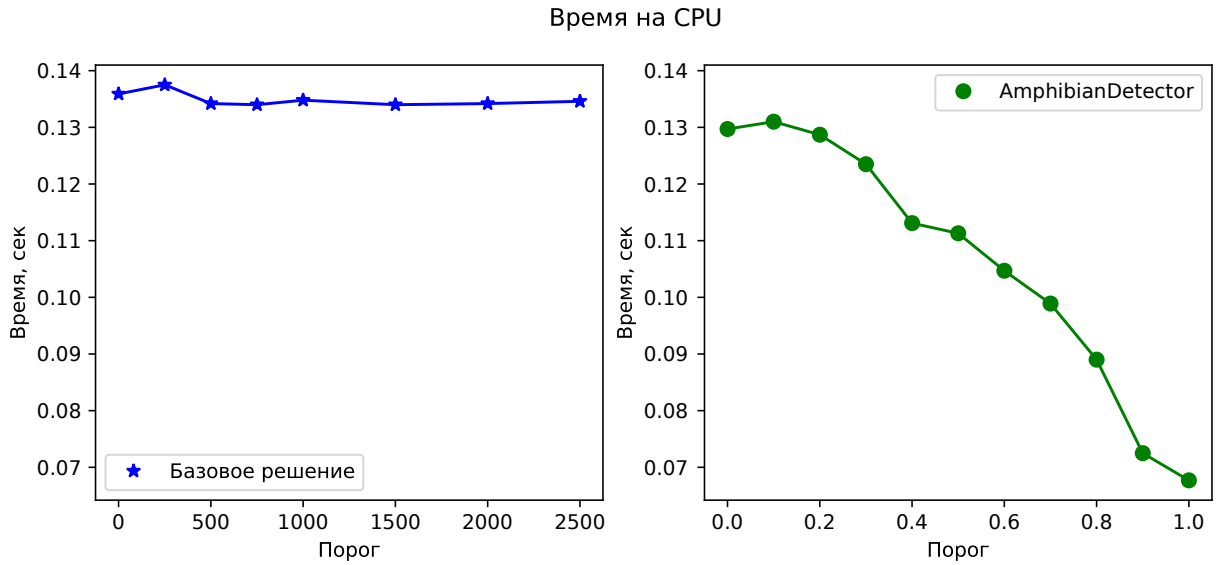

Рисунок 2.6 — Сравнение среднего времени обработки кадра для CDNet2014 pedestrian на Intel Core i5-4210U CPU 1.70GHz × 4 для базового подхода и AmphibianDetector на основе SSD+MobileNetV2.

Для базового решения значение порога варьировалось от 0 до 2500 с шагом 500, так как для большего значения порога mAP уменьшался до 0. Замеры AmphibianDetector с MobileNetV2+SSD проводились для номера слоя $m = 11$, как показавшего наилучшее соотношение mAP и времени для поднабора из 2000 кадров. На полученных графиках видно, что mAP для AmphibianDetector выше mAP для базового решения почти на 15% (Рис. 2.5) при выборе лучшего порога для каждого подхода. Базовое решение достигает 0.405 mAP для порога равного 500. AmphibianDetector достигает 0.548 для порога равного 0.4. Стоит отметить, что в приведённых экспериментах максимально достижимый mAP ограничен точностью SSD+MobileNetv2 для рассматриваемых данных.

Предложенный подход AmphibianDetector позволяет снизить среднее время обработки кадра. Для сравнения среднего времени обработки кадра были выполнены замеры времени на Intel Core i5-4210U CPU 1.70GHz × 4. Значение порогаварьировалось в тех же диапазонах что и для замеров mAP. Из графиков видно (Рис. 2.6), что AmphibianDetector позволяет сократить среднее время на обработку кадра за счёт эффективного отфильтровывания кадров без движе-

Таблица 3 — Сравнение mean average precision для SSD с извлекателем признаков MobileNetV2 и AmphibianDtector на основе SSD+MobileNetV2.

Модель	mAP
SSD+MobileNetV2	0.326
AmphibianDetector (MNV2; m=11, λ=0.4)	0.548

ния, что достигается использованием для сравнения кадров карт признаков из промежуточного слоя сети.

Было проведено сравнение точности AmphibianDetector на основе SSD+MobileNetV2 со стандартным использованием SSD+MobileNetV2. Значения карт признаков брались со слоя $m = 11$. Пороговое значение λ задавалось равным 0.4. Для сравнения точности мы сравнивали значение mAP для стандартного использования детектора и AmphibianDetector. SSD+MobileNetV2 позволяет достигать mAP равного 0.326, применение AmphibianDetector позволяет увеличить mAP до 0.548.

2.6 Заключение по второй главе

В данной главе был предложен подход к ускорению работы детектора за счёт отфильтровывания из видео потока кадров, не содержащих движущихся объектов. Было экспериментально показано, что предложенный подход уменьшает среднее время обработки кадра по сравнению с базовым подходом на основе попиксельного сравнения кадров. Предложенный подход позволяет сократить число ложных срабатываний сети, что может быть критично в ряде прикладных задач. Таких как, например, распознавание лиц, где важно не допустить подачи некорректной области изображения в модель, производящую идентификацию. Исходный код с реализацией предложенного подхода и экспериментами доступен в открытом доступе на GitHub[94].

Глава 3. Оптимизация вычисления биометрического вектора

3.1 Функция Софтмакс в задаче распознавания лиц

Функция Софтмакс широко используется при обучении нейронных сетей для решения задач классификации [1—3]. СНС для классификации состоят из двух составных частей: извлекателя признаков и головы-классификатора. Извлекатель признаков, последовательным применением операций свёртки с обучаемыми фильтрами и операций понижения размерности, преобразует входное изображение в вектор признаков. Голова-классификатор на основе полученного вектора возвращает условные вероятности соответствия заданным классам объектов. Для этого голова-классификатор преобразует вектор признаков в выходной вектор модели умножением на обучаемую матрицу весов нейронной сети. Количество элементов в выходном векторе соответствует числу классов по которым проводится классификация.

Для получения на выходе СНС значений, удовлетворяющих определению вероятности, используется функция Софтмакс (англ. Softmax), преобразующая каждое значение z_i в выходном векторе z следующим образом:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}},$$

где K задаёт количество возможных классов объектов. После такого преобразования все элементы вектора z имеют значения в интервале $[0, 1]$ и их сумма равна 1. Это позволяет обучать значения весов модели используя вероятностные методы, такие как метод максимального правдоподобия [95].

СНС для построения биометрических векторов обучается как задача классификации большого количества людей. После обучения у модели удаляется голова-классификатор и вектора признаков интерпретируются как биометрические вектора. На этапе обучения модели используется специальная модификация головы-классификатора для предания векторам признаков необходимых свойств:

– все вектора должны находиться на гиперсфере единичного радиуса;
– близость векторов определяется как угол между ними;
– близость векторов должна быть пропорциональна схожести соответствующих им лиц.

Для достижения данных свойств вектор признаков и значения матрицы весов нормируются к 1, а их произведение интерпретируется как косинус угла между векторами. Подробное рассмотрение данной модификации головы-классификатора приводится в разделе 3.4.

3.2 Задача распознавания лиц

Как продемонстрированно в предыдущих работах [52—54] использование СНС в сочетании с функцией Софтмакс с отступами (приводится в разделе 3.3) позволяет достичь наибольшей точности в задаче распознавания лиц. Развитие встраиваемых систем, таких как умные домофоны, породило интерес к легковесным нейронным сетям. Так были предложены облегченные нейросетевые модели, обученные с применением функции Софтмакс с отступами, для задачи идентификации по лицу. В данной диссертации предлагается метод дистилляции, который позволяет получить большую точность, чем другие методы для задачи распознавания лиц на наборах данных LFW, AgeDB-30 и Megaface. Основная идея предлагаемого подхода заключается в использовании центров классов сети-учителя для инициализации сети-ученика. Затем сеть-ученик обучается производить биометрические вектора, углы от которых до центров классов равны углам в сети-учителе.

Встраиваемые системы распознавания лиц основываются на нейросетевых моделях для мобильных вычислителей. Во время работы нейросетевая модель получает на вход изображение лица и восстанавливает по нему вектор фиксированной длины. В таком векторе закодирована информация о лице на изображении, и он называется биометрическим. Чем дальше вектора для различных людей друг от друга в векторном пространстве и ближе для различных

изображений одного человека, тем выше качество работы нейронной сети. Близость векторов определяется согласно выбранной метрике, в рассматриваемых в данной работе моделях это косинусное расстояние.

К таким моделям относится архитектура MobileFaceNet [21], разработанная специально для распознавания лиц на устройствах с малой вычислительной мощностью. В свою очередь использование функции Софтмакс с отступами [52—54] для обучения таких моделей позволяет достичь наибольшую точность в задачах идентификации и верификации по лицу. Нейросетевые модели для биоидентификации обучаются на классификацию с функцией Софтмакс большого количества классов, затем вектор, получаемый на предпоследнем слое, используется для кодирования и сравнения новых изображений. Добавление отступов в функцию Софтмакс в процессе обучения сети добавляет дополнительный отступ для векторов, относящихся к одному классу, что вынуждает сеть минимизировать угол от вектора до центра класса. Таким образом сеть обучается минимизировать косинусное расстояние между изображениями одного человека (Рис. 3.1).

Быстрые и компактные мобильные нейросетевые архитектуры достигают меньшей точности, чем серверные решения. В задачах биометрического доступа такое снижение точности может играть критическую роль. Для увеличения точности мобильных нейросетевых архитектур используется дистилляция [26]. Дистилляция нейронной сети — это метод передачи знаний от сети-учителя с большим числом обучаемых параметров в сеть-ученика с малым числом обучаемых параметров. В данной диссертации предлагается новый подход дистилляции, позволяющий сократить разницу в точности между сетью-учителем и сетью-учеником.

Идея предложенного подхода заключается в копировании последнего слоя сети, содержащего обученные центры классов обучающего набора данных, из сети-учителя в сеть-ученика и заморозке этого слоя во время всей процедуры дистилляции. Дистилляция заключается в обучении сети-ученика воспроизводить углы между центрами классов и биометрическими векторами для лиц, равные углам между соответствующими векторами и центрами классов в сети-

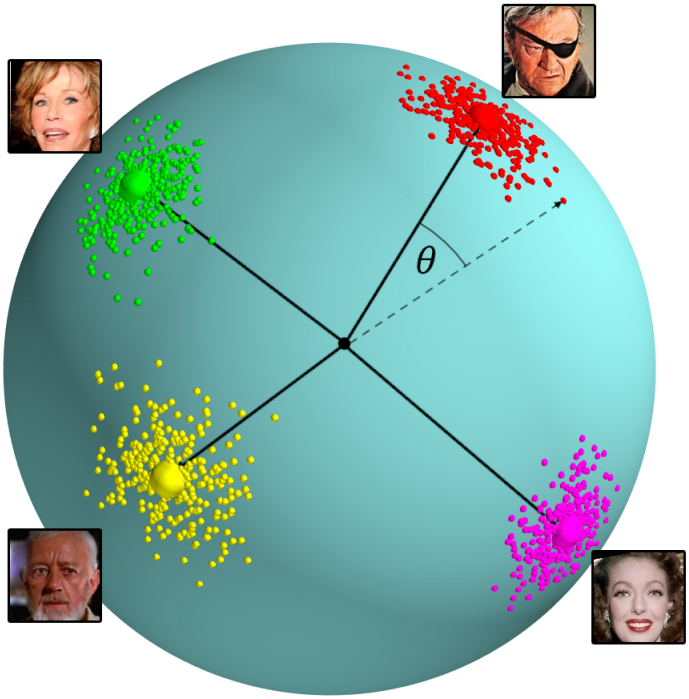

Рисунок 3.1 — Гиперсфера нормированных биометрических векторов. Различными цветами обозначены разные классы из обучающего набора данных: большими кругами обозначены центры классов, меньшими - изображения, относящиеся к классу. Для вычисления близости изображения к классу необходимо вычислить угол θ между векторами.

учителе. Такой подход позволяет сети-ученику лучше воспроизводить результат работы сети-учителя.

3.3 Существующие подходы к построению биометрических векторов

Функция Софтмакс с отступами

Существует несколько вариаций функции Софтмакс (анг. softmax) с отступами, используемых при обучении нейронных сетей для систем распознавания лиц. Они включают подходы Cosface [52], Sphereface [53] и Arcface [54]. Все три подхода могут быть описаны общей формулой, задающей функцию ошибки классификации изображений лиц:

$$L = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s(cos(\theta_{y_i}m_1+m_2)-m_3)}}{e^{s(cos(\theta_{y_i}m_1+m_2)-m_3))} + \sum_{j=1,j\neq y_i}^{n} e^{s\,cos\theta_j}}.$$

Данные подходы могут быть получены из представленной формулы подстановкой значений параметров m_1, m_2, m_3. Подход Sphereface получается при $m_1 = 4, m_2 = m_3 = 0$; подход Cosface получается при $m_1 = 1, m_2 = 0, m_3 = 0.35$; подход Arcface получается при $m_1 = 1, m_2 = 0.5, m_3 = 0$. В представленной формуле N - это количество изображений, используемых на каждом шаге стохастического градиентного спуска; θ_j - угол между векторами, соответствующими обучающему примеру с индексом i и центру класса с индексом j; y_i - соответствует индексу класса, к которому относится пример с индексом i согласно разметке; s - константа масштабирования, во всех случаях имеющая значение 64. Детальный анализ выбранных значений параметров представленной формулы можно найти в статье Arcface [54]. Из рассмотренных подходов Arcface демонстрирует наибольшую точность на наборах данных LFW, AgeDB-30 и MegaFace.

Дистилляция

Дистилляция знаний от сети-учителя к сети-ученику была впервые предложена в работе [26]. Данный подход заключается в обучении сети-ученика с малым числом обучаемых параметров за счёт передачи знаний от сети-учителя с большим числом обучаемых параметров. Под передачей знаний понимается передача некоторой информации от обученной более точной модели для улучшения сходимости обучаемой малой модели. Ключевая идея подхода, предложенного в работе [26], заключается в передаче знаний через приближение сглаженного вероятностного распределения на выходе сети-учителя. Это достигалось за счёт добавления делителя в формулу функции Софтмакс.

Некоторые исследования продолжают развивать идею использования сглаженного распределения вероятности в качестве разметки для обучения сети-ученика. Так, например, в работе [45] был предложен подход к дистилляции ансамбля нейронных сетей в одну сеть-ученика, используя данный метод. В работе [47] предложен подход к обучению сети-ученика с использованием регуляризации на основе добавления шума к ответам учителя, имитируя таким

образом использование нескольких учителей. В работе [48] сеть-ученик и сеть-учитель обучались с одинаковой параметризацией. В работе [96] предлагается механизм дистилляции через задание априорного распределения сети-ученика на основе апостериорного распределения сети-учителя с изменённой структурой модели для совпадения пространства параметров. Также делаются шаги в сторону теоретического обоснования методов дистилляции через вероятностную интерпретацию [96; 97].

Другой подход к дистилляции — это дистилляция через скрытые слои нейронной сети. В [98] предлагается обучать сеть-ученика копировать распределение весов на скрытых слоях сети-учителя. В [49] промежуточные слои сетей ученика и учителя используются для регуляризации обучения. В [50] для регуляризации во время дистилляции используется ограничение на сохранение взаимоотношения между локальными объектами. Для этого расстояние между биометрическими векторами для сети-ученика минимизируется на основе информации от сети-учителя. В [51] предлагается относительная дистилляция знаний, которая штрафует за структурные различия во взаимоотношении между обучающими примерами.

Для дистилляции легковесных нейронных сетей для задачи распознавания лиц, обученных с применением функции Софтмакс с отступами, применяются следующие подходы: триплетная дистилляция [55], дистилляция по углу [56] и дистилляция на основе отступа [57].

В *триплетной дистилляции* сеть-ученик обучается с триплетной функцией ошибки, отступ в которой вычисляется, основываясь на расстоянии между якорным и негативным примерами и якорным и позитивным примерами, предсказанными сетью-учителем.

В *дистилляции по углу* минимизируется угол между биометрическими векторами сетей ученика и учителя для каждого примера в обучающей выборке.

В *дистилляции на основе отступа* предлагается производить дистилляцию через сглаженное распределение вероятности. Для этого в формулу 1 добавляется деление на параметр T по аналогии с [26].

Таблица 4 — Сравнение параметров рассматриваемых нейросетевых архитектур.

	ResNet100	MobileFaceNet
Число операций с плавающей точкой / 10^9	24.2	0.44
Размер / Мегабайт	261.2	5.3
Число обучаемых параметров / 10^6	52.56	1.19
Время работы / Миллисекунд	401 ± 25.7	42.2 ± 5.48

3.4 Разработка специализированного метода дистилляции для задачи распознавания лиц

Описание сети-учителя и сети-ученика

В качестве сети-учителя рассматривалась нейронная сеть с архитектурой ResNet100 [15]. Выбор данной архитектуры был обусловлен тем, что она содержит большое число обучаемых параметров и, следовательно, позволяет достигать высокой точности в задаче распознавания лиц. В качестве сети-ученика была выбрана недавно предложенная архитектура MobileFaceNet(ReLU) [21], содержащая менее 1 миллиона параметров и разработанная специально для решения задачи распознавания лиц на мобильных процессорах. Данная архитектура состоит из блоков предложенных в MobileNetV2 [92], но позволяет обрабатывать изображения вдвое быстрее за счёт уменьшения пространственной размерности изображения на ранних слоях и использования меньшего числа фильтров на промежуточных слоях блоков. В описываемых экспериментах была сделана следующая модификация данной архитектуры: размерность выходного биометрического вектора была увеличена с 256 до 512 элементов для совпадения с размерностью архитектуры ResNet100. В таблице 4 приводится сравнение рассматриваемых архитектур. Время работы сетей замерялось для входного изображения размерностью $112 \times 112 \times 3$ на процессоре Intel Xeon(R) CPU E3-1270 v3 @ 3.50GHz $\times$ 8.

Описание метода

Обозначим биометрический вектор сети-ученика для изображения с индексом i как $x_{S_i} \in R^D$, где D - размерность вектора. И обозначим через $x_{T_i} \in R^D$ биометрический вектор соответствующего изображения для сети-учителя. Матрицы весов последнего слоя для сетей ученика и учителя будем обозначать соответственно $W_S \in R^{D \times n}$ и $W_T \in R^{D \times n}$, где n - количество классов в обучающем наборе данных. Столбец матрицы с индексом j, соответствующий центру класса y_i, к которому относится изображение с индексом i из обучающего набора данных, обозначается как $W_{S_j} \in R^D$ для сети-ученика и $W_{T_j} \in R^D$ для сети-учителя.

Подходы, основанные на добавлении отступа m в функцию Софтмакс, производят нормировку столбцов матрицы весов и биометрических векторов к 1: $||W_j|| = 1$ и $||x_i|| = 1$, где W_j - это j-й столбец матрицы W, а x_i - это биометрический вектор, соответствующий i-му обучающему примеру. Такая нормировка позволяет рассматривать результаты произведений векторов на последнем слое сети как косинусные расстояния $cos(\theta_j)$ между биометрическими векторами и соответствующими центрами классов: $W_j{}^T x_i = ||W_j|| \cdot ||x_i|| cos(\theta_j) = cos(\theta_j)$ (Рис. 3.1), где θ_j - угол между векторами, соответствующими обучающему примеру с индексом i и центру класса с индексом j. Далее в качестве частного случая обучения с использованием функции Софтмакс с отступами рассматривается метод Arcface [54], задаваемый формулой подробно описанной в разделе 3.3:

$$L_{\text{ArcFace}} = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{s(cos(\theta_{y_i} + m))}}{e^{s(cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^{n} e^{s\, cos\theta_j}}.$$

Данный метод был выбран для рассмотрения, так как позволяет получать наибольшую точность среди методов обучения с применением функции Софтмакс с отступами. В подходе Arcface значение отступа m фиксируется равным 0.5, как предложено в [54]. В данной работе предлагается производить дистилляцию знаний от сети-учителя через вычисление значения отступа m для каждого изображения i.

Предлагаемый метод дистилляции содержит две основные идеи:

– Центры классов, найденные сетью-учителем, используются в сети-ученике: $W_S = W_T$. Так как центры классов обучаемые параметры, более глубокая сеть способна сформировать более хорошие их положение на гиперсфере. Такое положение, при котором относящиеся к этим центрам кластеры векторов будут более компактны и сильнее разнесены в пространстве.
– Для дистилляции знаний используются вычисляемые значения m_i для каждого изображения. Они явным образом контролируют расстояние между биометрическими векторами x_{S_i} и соответствующими центрами классов W_{S_j}. Большее значение m_i способствует приближению вектора x_{S_i} к центру класса.

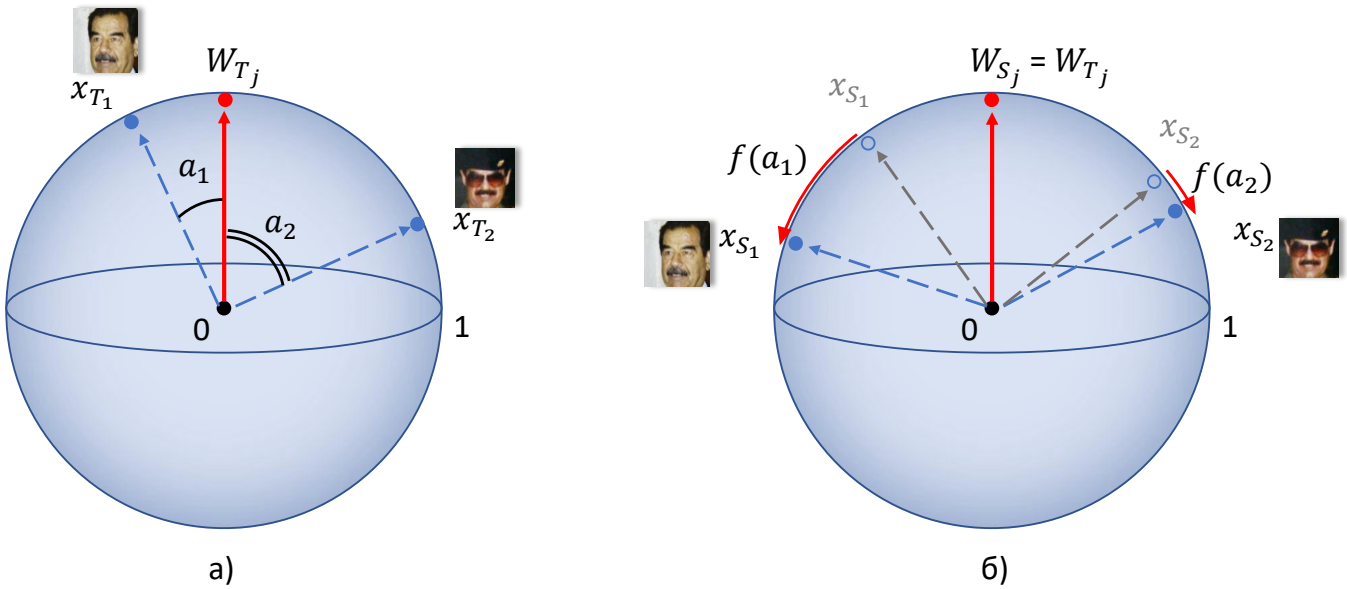

Рисунок 3.2 — а) Гиперсфера векторов сети-учителя. б) Вычисление сдвига для векторов на гиперсфере сети-ученика на основе углов до центра класса в сети-учителе.

Отступ m_i вычисляется основываясь на угле между биометрическим вектором сети-учителя $x_{T_i} \in R^D$ и соответствующим вектором $W_{T_j} \in R^D$ центра класса, задаваемого как y_i. Отступ m для изображения с индексом i вычисляется аналогично отступу в триплетной дистилляции для ограничения его максимальных и минимальных значений:

$$m_i = f(a_i) = \frac{m_{max} - m_{min}}{a_{max}} a_i + m_{min},$$

$$a_i = \frac{W_{T_j}^T x_{T_i}}{||W_{T_j}|| \cdot ||x_{T_i}||}, \tag{3.1}$$

где параметры $m_{max} = 0.5$ и $m_{min} = 0.2$ задают максимальное и минимальное значение отступа, по аналогии с формулой триплетной дистилляции предложенной в [55]. А значение параметра a_{max} вычисляется как максимальное значение угла a в мини-пакете изображений на каждом шаге обучения.

Формула 3.1 используется для вычисления углов a_i между центрами классов W_{T_j} и биометрическими векторами x_{T_i} сети-учителя (Рис. 3.2 (а)). Углы a_i, получаемые от сети-учителя, используются для вычисления отступа $m_i = f(a_i)$ по формуле выше. Чем меньше значение угла между биометрическим вектором и центром класса к которому он относится, тем больший отступ m_i используется для этого вектора при обучении сети-ученика (Рис. 3.2 (б)). Большее значение отступа вынуждает сеть минимизировать угол от биометрического вектора до центра класса, чтобы компенсировать эффект от отступа. Тем самым векторы, которые были близки к центру класса в сети-учителе, будут близки к центру класса в сети-ученике.

Интуиция, лежащая в основе предлагаемого подхода, заключается в том, чтобы изменять биометрический вектор сети-ученика для уменьшения расстояния до центра класса, если соответствующий вектора сети-учителя находится близко к центру класса. Это позволяет осуществлять передачу знаний от сети-учителя к сети-ученику более эффективно потому, что сеть-ученик концентрируется на более уверенно классифицированных сетью-учителем изображениях.

Предлагаемый подход к дистилляции позволяет передавать информацию об относительном положении векторов на гиперсфере, не вводя явных ограничений на биометрические вектора сети-ученика.

3.5 Экспериментальная апробация предлагаемого метода

Детали реализации

Предобработка данных. Для обнаружения и выравнивания лиц на изображениях применялся метод MTCNN [99]. Выравнивание осуществалялось аффинной трансформацией изображения лица с матрицой трансформации, вычесленной для позиционирования ключевых точек. Ключевые точки соотвествуют положению глаз, носа и углов рта и при выравнивании должны быть перемещены в заданные позиции. В качестве обучающего набора данных использовался набор данных для классификации по лицу MS1MV2 [54]. Данный набор данных — это очищенный в полуавтоматическом режиме набор данных MS-Celeb-1M [100], предложенный в работе посвящённой Arcface. Рассматриваемый набор данных содержит 5.8 миллионов изображений для 85 тысяч классов.

После процедуры выравнивания лиц, выполненной с применением ключевых точек найденных MTCNN, получаются изображения размером 112x112 пикселей. Значения яркости пикселей полученных изображений затем нормируются до диапазона [-1, 1].

Процедура обучения. В качестве сети-учителя была обучена нейронная сеть с архитектурой ResNet100 и функцией Arcface. Сравнение всех методов дистилляции проводилось в одинаковых условиях, где передача знаний осуществлялась от ResNet100 к MobileFaceNet(ReLU). Дистилляция производилась со следующими значениями гиперпараметров: размер мини-пакета изображений равнялся 512, шаг обучения равнялся 0.1 и увеличивался в 10 раз после 10'000, 160'000 и 200'000 итераций. Оптимизация обучаемых параметров выполнялась алгоритмом SGD [101] со значением моментов равным 0.9. Коэффициент регуляризации обучаемых параметров равнялся $5 * 10^{-4}$. Значения максимального и минимального возможных значений отступа фиксировались равными $m_{max} = 0.5$ и $m_{min} = 0.2$. Значение параметра s метода Arcface не изменялось и равнялось 64. Описанные далее эксперименты основываются на официальной реализации Arcface авторами на фреймворке MXNet.

Для сравнения предложенного метода дистилляции с существующими подходами, были реализованы триплетная дистилляция [55], дистилляция по углу [56] и дистилляция на основе отступа [57] на фреймворке MXNet. Нейронные сети дистиллировались с помощью этих методов с указанными в опорных статьях параметрами. Реализация этих методов на MXNet доступна в репозитории данной работы на GitHub[102].

Процедура тестирования. Обученные с помощью дистилляции нейронные сети для распознавания лиц тестировались на задачах верификации и идентификации.

Верификация. Для замеров точности обученных нейронных сетей на задаче верификации использовались наборы данных LFW и AgeDB-30. Каждый набор данных содержит порядка 3000 позитивных и 3000 негативных пар изображений. В процедуре тестирования обученная нейронная сеть использовалась для получения биометрических векторов для пары изображений лиц. Верификация производилась, основываясь на косинусном расстоянии между векторами. Точность измерялась как процент верно верифицированных пар изображений.

Идентификация. Наиболее представительным и сложным протоколом тестирования для задачи идентификации лиц является MegaFace. В набор данных MegaFace входит миллион изображений лиц для 690'000 человек для формирования негативных примеров. И 100'000 изображений для 530 людей из набора данных FaceScrub [103], идентификацию которых по базе лиц необходимо произвести. Значением метрики качества является процент верно идентифицированных по базе лиц с 1 миллионом негативных примеров.

Результаты тестирования

Как показано в таблице 5, обученная с функцией Arcface сеть-учитель достигает точности 99.76 % для набора данных LFW и 98.21 % для AgeDB-30. Сеть-ученик, обученная с функцией Arcface, достигает 99.51 % для набора данных LFW и 96.13% для AgeDB-30. Предложенный подход позволяет сократить разницу в точности сильнее, чем остальные рассмотренные подходы: до 99.61 % для LFW и 96.55 % для AgeDB-30.

Таблица 5 — Точность верификации на наборах данных LFW и AgeDB-30. В экспериментах использовалась версия MobileFaceNet с функцией активации ReLU.

Архитектура	Метод обучения	LFW %	AgeDB-30 %
ResNet100 (сеть-учителя)	ArcFace[54]	99.76	98.21
MobileFaceNet (сеть-ученик)	ArcFace[54]	99.51	96.13
MobileFaceNet	Триплетная дистилляция по L2[55]	99.56	96.23
MobileFaceNet	Триплетная дистилляция по cos[55]	99.55	95.60
MobileFaceNet	Дистилляция с отступом для T=4[57]	99.41	96.01
MobileFaceNet	Дистилляция по углу[56]	99.55	96.01
MobileFaceNet	Предлагаемый метод	**99.61**	**96.55**

Таблица 6 — Точность идентификации с использованием протокола MegaFace с 1 миллионом негативных примеров. В экспериментах использовалась версия MobileFaceNet с функцией активации ReLU.

Архитектура	Метод-обучения	MegaFace %
ResNet100 (сеть-учитель)	ArcFace[54]	98.35
MobileFaceNet (сеть-ученик)	ArcFace[54]	90.62
MobileFaceNet	Триплетная дистилляция по L2[55]	89.10
MobileFaceNet	Триплетная дистилляция по cos[55]	86.52
MobileFaceNet	Дистилляция с отступом для T=4[57]	90.77
MobileFaceNet	Дистилляция по углу[56]	90.73
MobileFaceNet	Предлагаемый метод	**91.70**

В таблице 6 представлены результаты замеров точности для задачи идентификации по протоколу MegaFace с 1 миллионом негативных примеров в базе лиц. Негативнае примеры добавляются в базу лиц вместе с изображениями людей, идентификацию которых необходимо произвести, для услажнения задачи идентификации по такой базе. Для данного протокола тестирования сеть-учитель достигает точности 98.35 %, а сеть-ученик 90.62 %. Предложенный подход позволяет увеличить точность идентификации до 91.70 %. Также для задачи идентификации было замечено уменьшение точности для метода триплетной дистилляция, но данный метод показал хорошую точность для задачи верификации.

Для более детального анализа предлагаемого подхода была проведена его оценка методом удаления различных шагов алгоритма. При удалении элементов подхода оценивалось их влияние на точность верификации на наборе данных LFW. В таблице 7 оценивается влияние следующих аспектов метода:

- Копирование центров - копирование центров классов, представленных обучаемыми параметрами последнего слоя нейронной сети, от сети-учителя к сети-ученику: $W_S = W_T$.
- Использование m_i вместо m - использование вычисляемых m_i для дистилляции через Arcface вместо фиксированного $m = 0.5$.
- Фиксирование центров - пометка центров классов сети-ученика W_S как необучаемые параметры. Чтобы в процессе дистилляции они оставались равными центрам классов сети-учителя W_T.

Наибольший прирост в точности в 0.9 % вызван копированием центров классов сети-учителя в сеть-ученика и дальнейшая пометка их как необучаемые параметры. Самостоятельное обучение сетью-учеником центров классов W_S во всех сценариях ведёт к уменьшению точности на наборе данных LFW. Соответственно ключевую роль в предложенном методе дистилляции играет копирование центров классов. Использование адаптивного отступа m_i позволяет дополнительно увеличить точность получаемой модели.

Таблица 7 — Оценка точности метода на наборе данных LFW удалением различных шагов алгоритма.

Копирование центров	Использование m_i вместо m	Фиксирование центров	LFW %
✓	✓	✓	**99.61**
✓	✓		98.31
✓			99.55
	✓		99.43
✓		✓	99.60

3.6 Заключение по третьей главе

В данной главе был предложен подход к дистилляции нейронных сетей, обученных для задачи распознавания лиц с функцией Софтмакс с отступами. Была продемонстрирована эффективность использования центров классов сети-учителя в сети-ученике. Было проведено сравнение предложенного метода с другими методами дистилляции для нейронных сетей использующих Софтмакс с отступами. Описанный в данной работе подход позволяет получить лучшую точность на наборах данных LFW и AgeDB-30 для задачи верификации и для MegaFace для задачи идентификации.

Предложенный метод дистилляции может применяться для увеличения точности нейросетевых моделей с малым числом параметров для встраиваемых устройств. Таких, например, как камеры наружного наблюдения или домофоны с функцией доступа по лицу.

Глава 4. Разработка программного модуля распознавания лиц для встраиваемых систем

В данной главе представлено описание разработки и реализации системы распознавания лиц в виде набора нейронных сетей, инструмента для их дистилляции и программного модуля для работы на борту маломощных встраиваемых устройств. К сценариям применения данной системы могут быть отнесены: домофон с функцией распознавания жильцов, система биометрической идентификации смартфона, пропускная система предприятия.

Разработку системы распознавания лиц можно разделить на три этапа, представленных на рисунке 4.1. А именно: 1) обучение всех необходимых для работы системы нейронных сетей; 2) конвертация моделей из формата, в котором они были обучены, в оптимальный для конечного вычислителя формат; 3) реализация программного модуля для их выполнения на конечном устройстве пользователя со всей необходимой обработкой входных и выходных данных.

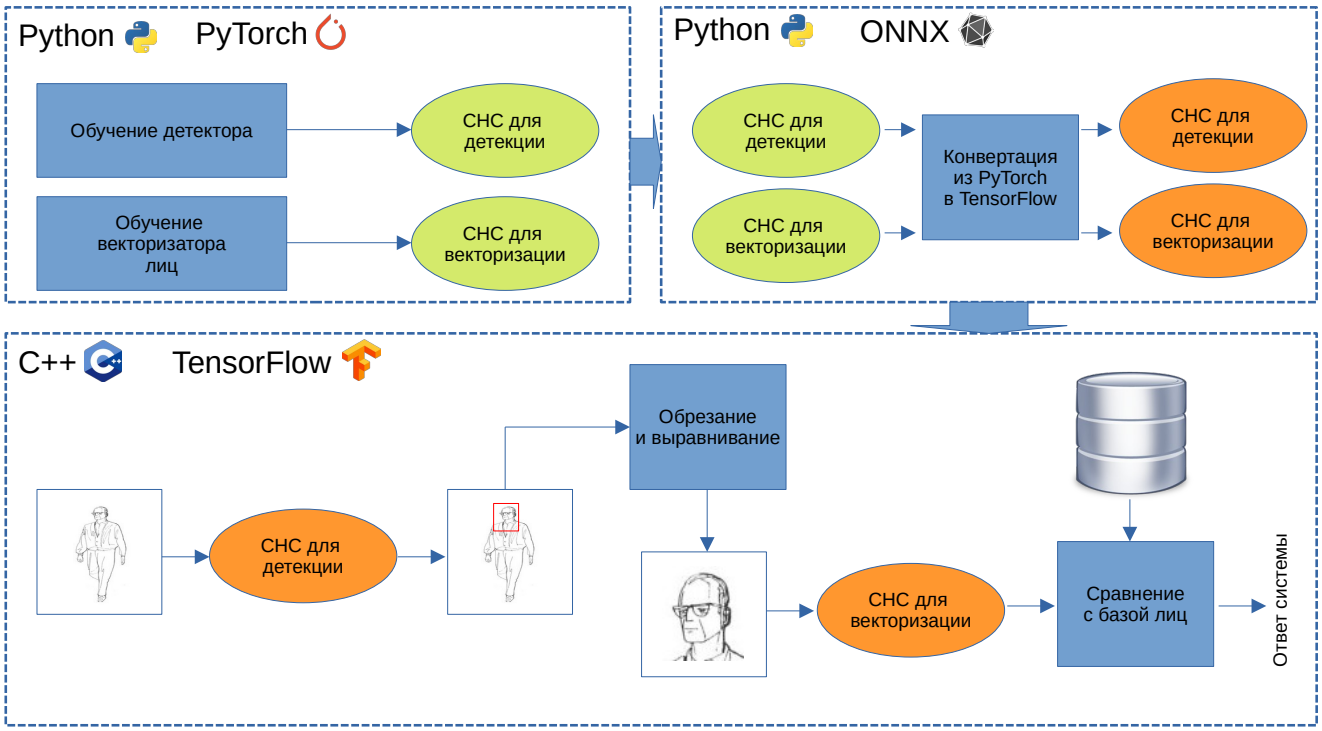

Рисунок 4.1 — Схематическое представление трёх этапов разработки системы распознавания лиц: обучение нейронных сетей, конвертация обученных параметров, разработка ключевых узлов программного модуля.

4.1 Обучение нейронных сетей

На этапе обучения нейронных сетей необходимо получить две нейронных сети: для обнаружения лица и ключевых точек на нём и для получения биометрического вектора по выравненному изображению лица.

В качестве нейронной сети для детекции лица и ключевых точек для получения наибольшей точности в серверных решениях используются нейронные сети с большим числом параметров, такие как RetinaFace [16]. Такая сеть позволяет одновременно обнаруживать на изображении лица в виде ограничивающих прямоугольников и ключевые точки (глаза, нос, рот), необходимые для надёжного выравнивания, на них. Однако такая модель требовательна к вычислительным ресурсам и может быть применена только для вычислений на сервере, предпочтительнее с графическим ускорителем.

Альтернативой использованию одной общей модели служит использование отдельных моделей для обнаружения лица и ключевых точек на нём. Так в представленной работе для обнаружения лица на маломощных вычислителях использовалась модель на основе архитектуры SSD [64] с извлекателем признаков MobilenetV2 [92]. Выбор данных архитектур обусловлен их вычислительной эффективностью и достаточно высокой точностью для решения задач биоидентификации. Данная модель была обучена с использованием фреймворка Object detection API с рекомендуемыми в оригинальной статье параметрами. В таком случае для решения задачи обнаружения ключевых точек на изображении (Рис. 4.2) используется СНС с малым числом параметров, решающая задачу регрессии для координат искомых точек.

Следующая нейронная сеть, которую необходимо обучить, - это нейронная сеть для построения биометрического вектора по выравненному изображению лица. Обучение такой сети производится в два этапа (Рис. 4.3). На первом этапе обучается нейронная сеть с большим числом параметров, такая как ResNet100 [15] с применением функции ошибки ArcFace [54]. Затем данная нейронная сеть используется для дистилляции с помощью описанного в третьей главе метода, а также может использоваться для вычисления биометрических векторов

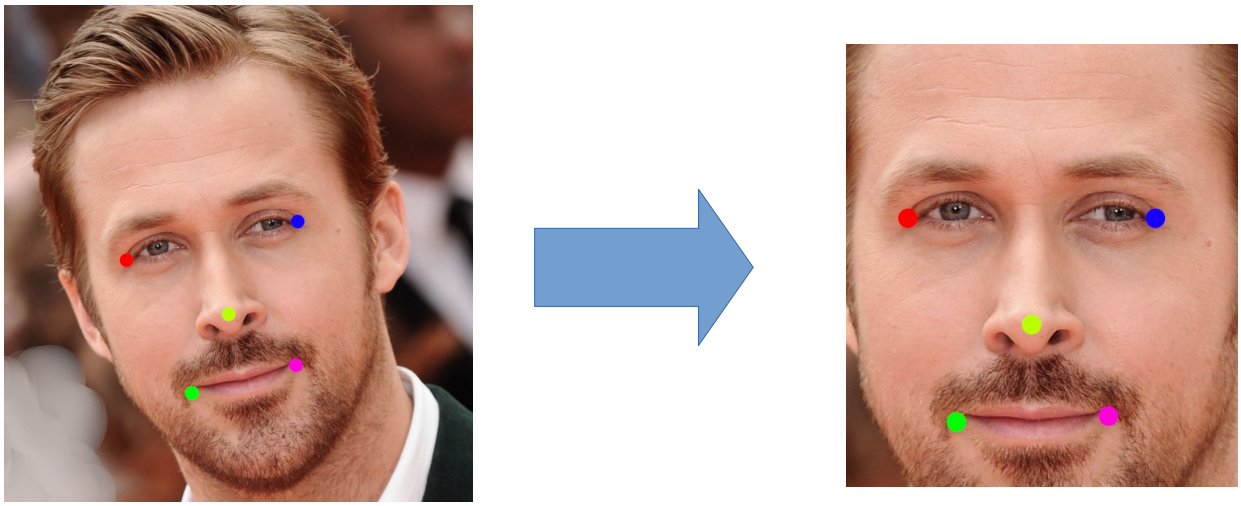

Рисунок 4.2 — Обнаружение ключевых точек (глаза, нос, рот) на изображении и применение аффинного преобразования для выравнивания по ним лица.

на стороне сервера. Дистилляция производится в сеть с малым числом параметров (Таблица 15), специально оптимизированную средствами NAS [32] для работы на маломощном устройстве. В данной работе использовалась архитектура сети MobiFace [22].

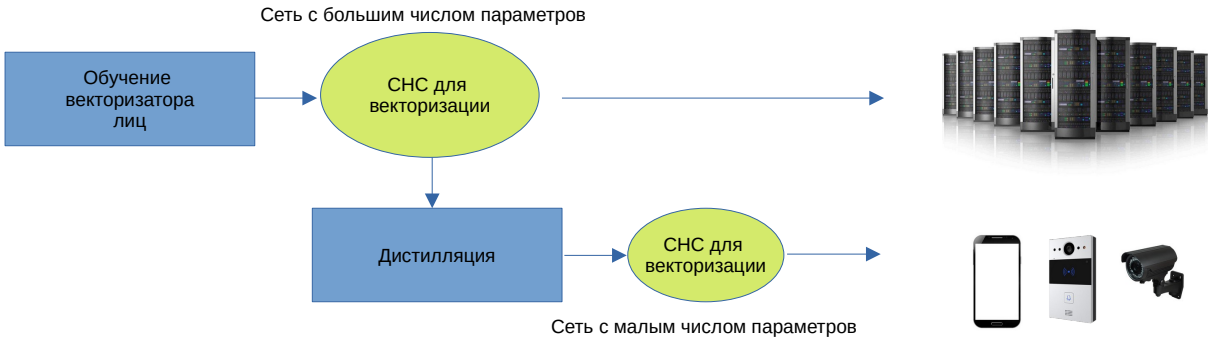

Рисунок 4.3 — Процедура обучения и дистилляции нейронных сетей для построения биометрического вектора лица.

Нейронные сети для построения биометрических векторов обучались с помощью фреймворка Pytorch [44], так как он упрощает отладку процесса обучения из-за организации нейронной сети в виде динамического графа.

4.2 Конвертация обученных параметров

На этапе обучения нейронной сети для вычисления биометрических векторов использовался фреймворк Pytorch, позволяющий оперировать нейронной сетью как динамическим вычислительным графом. Динамический вычислительный граф строится по мере обработки входных данных различными слоями нейронной сети и позволяет легко отлаживать процесс обучения сети стандартными средствами отладки языка программирования Python.

Альтернативой динамическому вычислительному графу является статичный вычислительный граф как во фреймворке TensorFlow [43]. Такой вычислительный граф позволяет оптимизировать вычисления в нём заранее с целью более полного использования вычислительных ресурсов конечного вычислительного устройства.

Для снижения количества возможных ошибок и ускорения процесса разработки на этапе обучения использовался подход с динамическим вычислительным графом. Затем полученные файлы с архитектурой сети и её обученными параметрами конвертировались средствами библиотеки Open Neural Network Exchange (ONNX) в статический граф TensorFlow. ONNX — это открытая библиотека программного обеспечения позволяющая обмениваться нейронными сетями между различными фреймворками для обучения и исполнения. После конвертации в TensorFlow в зависимости от требований к скорости работы и мощности целевого устройства опционально выполнялось квантование методом, предложенным в [31], но процесс квантования не является предметом исследования данной диссертации и подробно в ней не рассматривается.

4.3 Разработка ключевых узлов программного модуля

Как видно из схемы на рисунке 4.1, система содержит два ключевых программных узла: "выравнивание лица на изображении"и "сравнение с базой данных биометрических векторов".

Первый узел реализует выравнивание лица по ключевым точкам, как представлено на рисунке 4.2. По обнаруженным ключевым точкам и их каноничному заранее известному положению на обрезанном изображении фиксированного размера (112x112 пикселей) строится матрица трансформации. К пикселям входного изображения применяется аффинное преобразование с параметрами этой матрицы. В результате такой трансформации - ключевые точки изображения переносятся в их целевые позиции, при этом приводя изображение лица к стандартизированному виду. Это необходимый шаг обработки входных данных, так как стандартизация данных на входе нейронной сети позволяет существенно увеличить её точность.

Другим ключевым узлом программной системы является сравнение векторов с базой данных пользователей системы. В данном случае база данных хранится в виде обезличенного массива векторов, хранящихся на сервере или на самом устройстве, и представляющих собой биометрические вектора людей, имеющих доступ к объекту.

Также программным модулем является предложенный во второй главе данной диссертации метод снижения числа ложных срабатываний детектора. Так как предложенный метод не требует обучения дополнительных нейронных сетей, он реализуется в виде программной манипуляции над уже обученной нейронной сетью для детекции.

Все описанные программные модули были реализованы в двух вариантах на языках программирования: Python и C++. Первый вариант, на Python, используется для внедрения серверных решений и позволяет осуществлять более простую интеграцию с готовыми серверными решениями заказчика и более простую настройку эффективного использования вычислительных ресурсов. Реализация на C++ используется для внедрения в виде программного модуля в операционную систему маломощных вычислительных устройств таких как домофоны или смартфоны с функцией распознавания лиц.

4.4 Тестирование системы институтом стандартов NIST

При внедрении разработанной системы было проведено комплексное тестирование её реализации на C++ Американским национальным институтом стандартов (NIST) [104].

В рамках данного тестирования проверялась точность системы распознавания лиц как единого механизма, где на вход подаётся пара изображений, содержащая людей, а на выход выдаётся ответ об идентичности представленных людей.

Для тестирования NIST использует три различных набора данных (Рис. 4.4): Visa и Border - лабораторные данные, собранные в контролируемых условиях фотографии людей при оформлении визы и посадке на самолёт; Wild - собранные в неконтролируемых условиях данные с камер наружного наблюдения. Каждый набор данных содержит порядка 10^5 изображений. В качестве метрики использовалась метрика FNMR (False non-match rates: доля неверно отмеченных как различные люди пар изображений) при фиксированном FMR (False match rate: доля неверно отмеченных как одинаковые люди пар изображений). Формула для вычисления FNMR имеет следующий вид:

$$FNMR(T) = 1 - \frac{1}{N}\sum_{i=1}^{N} H(u_i - T),$$

где N — это число пар изображений, содержащих одного человека, в тестовом наборе данных, u_i - косинусное расстояние между биометрическими векторами пар изображений одного человека, T - пороговое значение, выбираемое на основе FMR, и H - ступенчатая функция:

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geqslant 0 \end{cases}$$

Значение порога T выбирается таким образом, чтобы значение FMR максимально совпадало с целевым заданным значением:

$$FMR(T) = \frac{1}{M}\sum_{i=1}^{M} H(v_i - T),$$

где M и v_i - задают число и косинусное расстояние для примеров пар изображений, содержащих различных людей.

Результат замеров точности предложенной системы приведён в таблице 8, полная таблица доступна в отчёте NIST [104].

Рисунок 4.4 — Пример изображений человека из наборов данных: а) Visa, б) Wild, в) Border

В тестируемой в NIST системе распознавания для получения наилучшего соотношения скорости и точности системы для построения биометрических векторов использовалась вариация архитектуры MobiFace [22] с увеличенным втрое числом свёрточных фильтров в каждом слое. Особенность данной архитектуры, позволяющая достигать ей высокой точности при сравнительно малом времени работы (Таблица 15), — это резкое понижение пространственного разрешения изображения на ранних слоях модели для более эффективной обработки данных. Такое понижение размерности позволяет увеличивать число свёрточных фильтров без значительного снижения скорости работы модели. В используемой в данной работе модели вместо функций активаций PReLU были использованы более современные Swish функции [105] для дальнейшего увеличения точности модели. Также после поздних свёрточных слоёв были добавлены se-блоки [106], позволяющие сети концентрироваться на определённых признаках изображения.

Таблица 8 — Сравнение FNMR для фиксированного FMR для различных наборов данных.

	False non-match rate (FNMR)		
Данные	Visa	Border	Wild
FMR	1E-06	1E-06	0.0001
FNMR	0.0274	0.9996	0.0337

Таблица 9 — Сравнение времени работы системы для различного разрешения входного изображения на CPU: Intel Xeon CPU E5-2630 v4 2.20GHz.

Разрешение	480x720	960x1440	1600x2400	3000x4500
Время (мс)	34	34	34	34

Согласно отчёту NIST реализация метода на языке C++ выполняется за фиксированное время в независимости от размера входного изображения (Табл. 9). Это обусловлено тем, что выбранная архитектура детектора на первом шаге уменьшает размер входного изображения до фиксированного, что не сказывается на точности потому, что на этапе выравнивания аффинное преобразование применяется к исходному изображению. Выполненная в рамках данной диссертации реализация демонстрирует одно из лучших соотношений скорости и точности работы системы.

Согласно проведённому институтом стандартов NIST сравнению с существующими системами, разработанная в рамках работы над диссертацией, система показала сравнимое качество на наборе данных Wild при существенно более низком времени отклика (Таблица 10). Однако была также отмечена просадка точности на наборах данных Border и Visa, так как обработка изображений лиц в высоком разрешении не является целевой задачей для разработанной системы.

Таблица 10 — Сравнение разработанной системы с решениями крупных российских вендоров технологии распознавания лиц на наборе данных Wild

Метод	FNMR @ FMR=0.0001	Время, мс
NtechLab	0.0292	1326
VisionLabs	0.0282	739
Предложенный метод	0.0337	34

4.5 Тестирование системы на данных с камеры домофона

Оценка точности работы детектора лиц

Для оценки качества работы разработанной системы в реальных условиях её применения было проведено тестирование точности на данных, собранных с целевых устройств. В качестве таких устройств выступали домофоны с видеокамерой, предоставленные компаниями ООО "Новотелеком"и ООО "Рубитек".

Оценка точности работы системы производилась в два этапа: сначала оценивалась точность работы детектора лиц, затем точность идентификации на основе биометрических векторов. Оценка точности работы детектора лиц оценивалась как полнота и точность при различных значения порога детектирования на реальных данных, собранных на камеру домофона, где полнота — это процент обнаруженных лиц из списка имеющихся в данных, а точность - процент того, сколько срабатываний детектора при заданном пороге действительно являются лицами.

Набор данных для тестирования качества детекции представляет собой 3730 кадров, содержащих изображения лиц, и 3467 кадров без лиц. Разметка кадров производилась в полуавтоматическом режиме - серверная нейронная сеть высокой точности отмечала на изображениях координаты лиц, которые затем отфильтровывались и уточнялись человеком.

Для замера полноты и точности детектора вводится критерий обнаружения лица детектора. Лицо считается обнаруженным, если при заданном пороге IoU превышает 50%, где IoU - широко используемая метрика совпадения огра-

ничивающих прямоугольников, равная отношению площади их пересечения к площади их объединения.

Используемый в системе детектор, описанный ранее в этом разделе, был протестирован на собранных данных. Для оценки свойств работы детектора была построена PR-кривая (Рис. 4.5), показывающая отношение точности и полноты при различных значениях порога. По динамике изменения кривой можно видеть, что детектор на целевых данных показывает высокую точность при значениях порога, дающих высокую полноту.

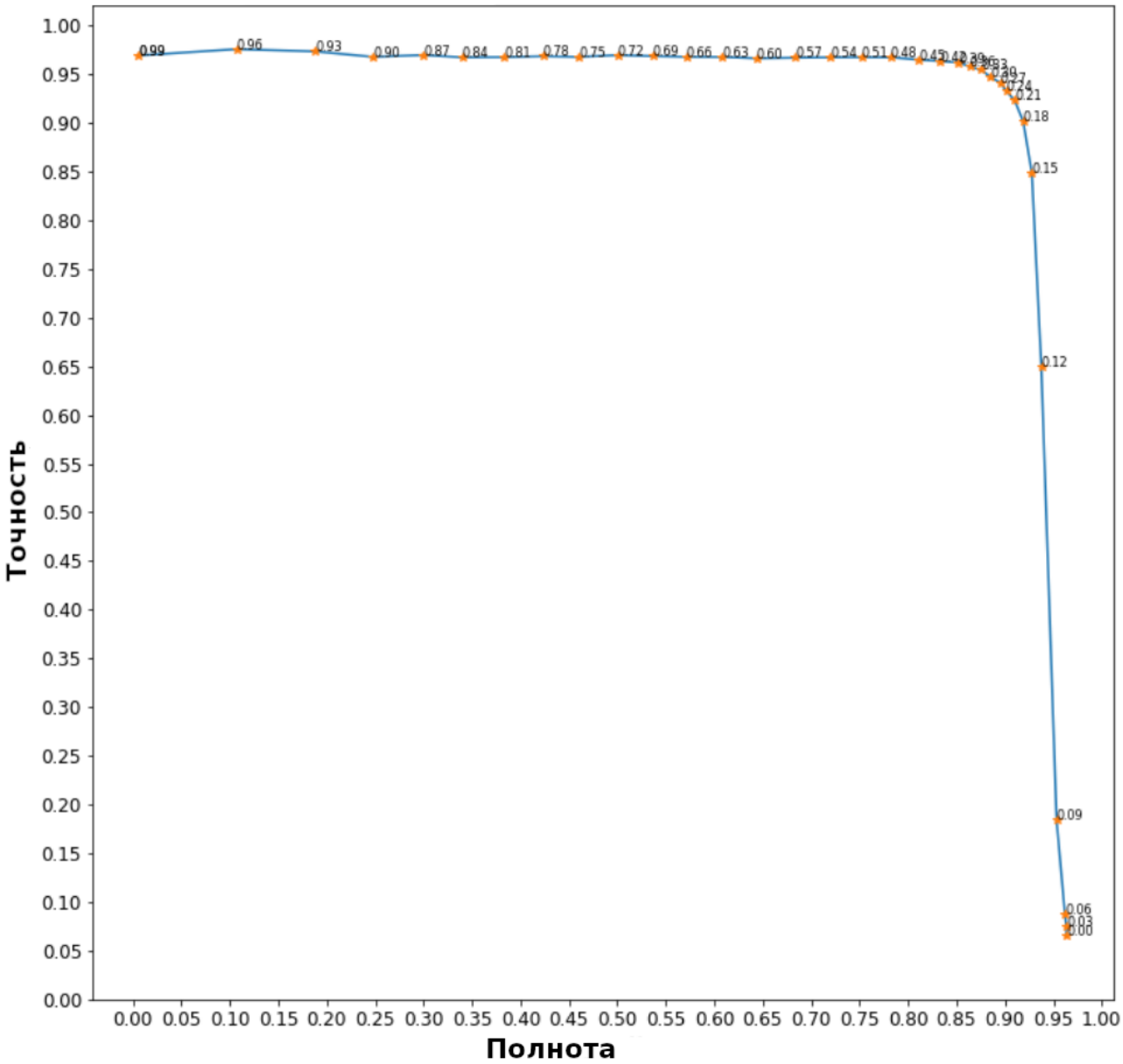

Рисунок 4.5 — PR-кривая, показывающая полноту и точность при различных значениях порога детектора лиц.

Таблица 11 — Точность дистиллированной модели после квантования и дообучения методом [58].

Модель	Точность на MegaFace
Resnet100 (учитель)	98.35%
MobiFace_x3_swish_se (ученик)	95.65%
MobiFace_x3_swish_se (дистиллированная)	96.35%
MobiFace_x3_swish_se (дистиллированная; квантованная)	96.03%

Оценка точности работы системы

В представленных в этом разделе экспериментах оценивалась на данных различного качества точность работы свёрточной нейронной сети для построения биометрических векторов. Далее нейронная сеть для детекции лиц оставалась зафиксированной для всех экспериментов. В том числе проводилось сравнение точности работы нейронной сети и её квантованной версии, так как при внедрении на конечном устройстве использовалась технология квантования весов нейронной сети до 8 бит [58]. Сравнение точности исходной и квантованных нейронных сетей на открытом наборе данных MegaFace представлено в таблице 11. Тестирование произведено в соответствии с методологией MegaFace. Модель MobiFace_x3_swish_se соответствует описанной в разделе 4.4 архитектуре.

Точность работы системы распознавания лиц оценивалась на данных с различным уровнем артефактов. Уровень артефактов определялся степенью сжатия алгоритмом кодирования потокового видео ffmpeg. Набор данных для оценки точности работы нейронной сети состоял из 5095 изображений, собранных с устройства с сильной степенью сжатия и 3802 изображений со слабой степенью сжатия. Для 17 и 13 уникальных людей соответственно. Описание используемых данных представлено в таблице 12, где число положительных ви-

Таблица 12 — Описание данных, собранных с использованием камеры видеодомофона для тестирования системы.

Описание данных	Количество людей в базе	Положительных визитов	Отрицательных визитов
Дневные данные, слабое сжатие, камера №1	13	224	3578
Дневные данные, сильное сжатие, камера №2	17	222	4873
Дневные данные, слабое сжатие, камера №2	18	647	2790
Ночные данные, слабое сжатие, камера №2	18	682	108

зитов характеризует сколько раз к домофону подходили люди из базы, а число отрицательных визитов - сколько раз подходили люди, которых в базе нет.

В качестве метрики точности системы использовалась TAR (True access rate) с порогом, соответствующим фиксированному FAR (False access rate). Использовался следующий протокол замера точности: сначала в базу добавлялось по одной фотографии каждого человека, собранной в лабораторных условиях, и по 500 фотографий случайных людей для повышения сложности правильного распознавания и проверки системы на устойчивость. Затем для всех изображений порог уверенности распознавания выбирался таким образом, чтобы значение FAR ровнялось 10%:

$$FAR(T) = \frac{1}{M}\sum_{i=1}^{M} H(v_i - T),$$

где M - число отрицательных визитов, а v_i - максимальная уверенность распознавания при сравнении со всеми изображениями в базе. Другими

словами: порог выбирается таким образом, чтобы процент ложных допусков равнялся выбранному значению. После фиксирования значения порога T целевая метрика TAR вычисляется следующим образом:

$$TAR(T) = \frac{1}{N} \sum_{i=1}^{N} H(u_i - T),$$

где N - число положительных визитов, а u_i - уверенность распознавания при сравнении с профилем соответствующего человека в базе лиц.

Результаты замеров TAR при FAR фиксированном на уровне 10% приведены в таблице 4.5. Уровень FAR обусловлен требованиями заказчика к надёжности системы. Из представленных замеров сети-учителя Resnet100 видно, что снижение качества входных данных из-за сжатия оказывает значительное негативное влияние на качество распознавания (100% TAR снижается до 91.84% TAR). При этом использование легковесной архитектуры MobiFace_x3_swish_se, описанной в разделе 4.4, приводит к значительному понижению TAR в том числе на данных со слабым сжатием, несмотря на то, что данная архитектура специализирована для задач лицевой биометрии. Повысить точность распознавания позволяет, предложенный в 3 главе данной диссертации, метод дистилляции (увеличение TAR на 4.08% для сильного сжатия и на 3.22% для слабого). Из-за регуляризационного эффекта, оказываемого нейронной сетью меньшей ёмкости и специализации архитектуры для систем распознавания лиц, в полной мере раскрывающейся при дистилляции, полученная модель превосходит точность сети-учителя на данных с сильным сжатием. Таким образом, предложенный метод позволяет эффективно увеличивать точность работы на данных с сильным сжатием. В последней строке таблицы 13 продемонстрировано, что квантование получаемых нейронных сетей методом [58] не приводит к снижению TAR.

Практическое внедрение распознавания лиц в домофонах накладывает требование на работу системы в тёмное время суток, в том числе для камер с инфракрасной подсветкой. Для проверки точности работы системы в таких условиях, был собран набор данных с инфракрасной подсветкой на домофон, предоставляющий такую возможность (последняя строка в таблице 12). Для собранных данных также производился замер TAR при фиксированном FAR.

Таблица 13 — Точность дистиллированной модели после квантования и дообучения методом [58].

Модель	Сильное сжатие TAR@FAR 10%	Слабое сжатие TAR@FAR 10%
Resnet100 (учитель)	91.84%	100%
MobiFace_x3_swish_se (ученик)	89.8%	93.55%
MobiFace_x3_swish_se (дистиллированная)	93.88%	96.77%
MobiFace_x3_swish_se (дистиллированная; квантованная)	93.88%	96.77%

Из результатов замеров в таблице 14 можно видеть, что для камеры рассматриваемого домофона использование ночного режима приводит к существенной просадке TAR, что особенно заметно для легковесной архитектуры сети-ученика. Использование предложенного в главе 3 метода дистилляции позволяет эффективно сократить это отставание в точности. Также из данной таблицы видно, что использование метода квантования [58] приводит к просадке TAR для инфракрасного режима работы камеры.

При внедрении систем распознавания лиц в домофонах нередко прибегают к обработке изображений лиц на сервере. В таком случае важно оптимизировать время работы системы для снижения стоимости вычислительных ресурсов и более простого масштабирования системы. В таблице 15 приведено сравнение времени работы нейронной сети с большим числом параметров Resnet100 и легковесной модели, описанной в разделе 4.4. Видно, что более лёгкая модель позволяет получить ускорение в 13 раз, при этом с близкой к исходной модели или большей точностью как показано в таблицах 13 и 14.

Для того, чтобы обосновать выбор в качестве сети-учителя Resnet100 и продемонстрировать, что данная модель не является излишне параметризованной, было проведено сравнение с моделью архитектуры из этого же семейства, но с вдвое меньшим числом параметров - Resnet50. В таблице 18 продемон-

Таблица 14 — Сравнение точности работы модели в ночное и дневное время.

Модель	Ночные данные TAR@FAR 10%	Дневные данные TAR@FAR 10%
Resnet100 (учитель)	91.94%	95.52%
MobiFace_x3_swish_se (ученик)	78.74%	91.91%
MobiFace_x3_swish_se (дистиллированная)	89.00%	93.66%
MobiFace_x3_swish_se (дистиллированная; квантованная)	85.48%	93.20%

Таблица 15 — Сравнение числа параметров и времени работы на 1th Gen Intel(R) Core(TM) i7-11700K @ 3.60GHz.

Модель	Время, мс	Число параметров
Resnet100 (учитель)	509.4	$52.56 * 10^6$
MobiFace_x3_swish_se (ученик)	36.7	$9.11 * 10^6$

стрировано, что такая модель выполняется в два раза быстрее, но как видно из таблиц 16 и 17, проигрывает по точности. Таким образом, в качестве сети-учителя для дистилляции используется Resnet100, так как время работы сети-учителя не влияет на время работы конечной системы, а снижение числа параметров приводит к снижению точности системы на данных собранных с помощью целевого устройства.

Таблица 16 — Сравнение сети-учителя с моделью той же архитектуры и вдвое меньшей ёмкости. Для данных с различной степенью сжатия.

Модель	Сильное сжатие TAR@FAR 10%	Слабое сжатие TAR@FAR 10%
Resnet100	91.84%	100%
Resnet50	87.76%	96.77%

Таблица 17 — Сравнение сети-учителя с моделью той же архитектуры и вдвое меньшей ёмкости. Для ночных (инфракрасный спектр) и дневных данных (RGB).

Модель	Ночные данные TAR@FAR 10%	Дневные данные TAR@FAR 10%
Resnet100	91.94%	95.52%
Resnet50	82.84%	91.19%

Таблица 18 — Сравнение сети-учителя с моделью той же архитектуры и вдвое меньшей ёмкости. Время работы.

Модель	Время, мс
Resnet100	509.4
Resnet50	278.7

4.6 Заключение по четвёртой главе

В данной главе подробно описаны алгоритмы, используемые при реализации системы, основанной на предлагаемых в данной диссертации подходах, к оптимизации времени работы.

Приведены результаты исчерпывающих экспериментов на реальных данных, собранных с камеры домофона, и замеры времени на сервере обработки лиц. Из представленных экспериментов видно, что предлагаемые методы поз-

волили ускорить построение биометрических векторов в 13 раз, при получении близкой к исходной модели или большей (на некоторых категориях данных) точности. Повышение точности на данных, подверженных сильному сжатию, объясняется регуляризационным эффектом сети малой ёмкости. Ограниченное число параметров не даёт ей переобучиться, в то время как дистилляция позволяет модели извлекать более сложные паттерны из изображений лиц.

В главе также представлены результаты анализа разработанной системы американским институтом стандартов NIST. Из отчёта NIST видно, что система позволяет получить достаточно высокую точность на данных в неконтролируемых (реальных) условиях, при этом демонстрируя высокую скорость работы.

Заключение

В результате выполнения данной диссертационной работы достигнуты следующие результаты:

1. Предложен метод инициализации весов СНС с малым числом параметров для обучения сети на задачи распознавания лиц. Предложенный метод демонстрирует эффективность использования весов последнего слоя сети с большим числом параметров в извлекателе признаков для инициализации сети с малым числом параметров.
2. В работе продемонстрировано повышение точности СНС для распознавания лиц на примере модели с архитектурой MobileFaceNet на наборе данных LFW с 99.51% до 99.60% за счёт применения разработанного метода инициализации последнего слоя сети-ученика значениями весов сети сети-учителя.
3. Предложен метод дистилляции для повышения точности СНС с малым числом параметров, обученных для задачи распознавания лиц. Предложенный метод дистилляции учитывает специфику работы нейронных сетей, обученных с функцией Софтмакс с отступами.
4. На примере нейронной сети с архитектурой MobileFaceNet был продемонстрирован прирост точности на наборе данных MegaFace с 90.62% до 91.70% за счёт применения разработанного подхода. Предложенный подход к дистилляции позволяет получить большую точность модели, чем применение прочих существующих методов дистилляции, позволявших получить точность до 90.77% на этом наборе данных.
5. Предложен метод для снижения числа ложноположительных срабатываний детектора людей в видеопоток. Это позволило повысить надёжность всей системы распознавания лиц, за счёт исключения из дальнейшей обработки случаев, приводящих к недетерминированному поведению системы. Метод использует промежуточные карты признаков из СНС, извлекающей признаки в детекторе для устойчивого к шуму обнаружения движения. Также предложенный подход позволил

ускорить этап детектирования человека за счёт ранней остановки работы детектора на кадрах, не содержащих движущихся объектов.

6. Реализация разработанного метода обнаружения движения на основе промежуточных карт признаков позволяет снизить среднее время работы нейронной сети с 0.135 сек до 0.112 сек для модели с архитектурой SSD+MobileNetV2. При применении реализации разработанного подхода метрика Mean Average Precision возросла с 0.326 до 0.548 на наборе данных CDNet2014 pedestrian.
7. Предложенные в данной работе методы реализованы и внедрены в виде ключевых модулей в системы распознавания лиц следующих партнёров ООО Экспасофт: ООО Новотелеком, ООО Рубитек РУС, ООО Открытая мобильная платформа, о чём имеется соответствующий акт о внедрении, прикреплённый к данной диссертации.
8. Для данных с видео домофона разработанный программный комплекс, реализающий представленные подходы, позволил произвести ускорение системы в 13 раз без просадки точности для некоторых сценариев использования. Предложенные методы и все приведённые эксперименты реализованы на языке Python с использованием библиотеки Pytorch и их исходный код выложен в открытый доступ на платформе GitHub для того, чтобы на них могли опираться дальнейшие исследования.

Список литературы

1. Imagenet: A large-scale hierarchical image database [Текст] / J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei // 2009 IEEE conference on computer vision and pattern recognition. — Ieee. 2009. — С. 248—255.
2. Gradient-based learning applied to document recognition [Текст] / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. — 1998. — Т. 86, № 11. — С. 2278—2324.
3. Learning multiple layers of features from tiny images [Текст] / A. Krizhevsky, G. Hinton [и др.]. — 2009.
4. The pascal visual object classes (voc) challenge [Текст] / M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman // International journal of computer vision. — 2010. — Т. 88, № 2. — С. 303—338.
5. The open images dataset v4 [Текст] / A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov [и др.] // International Journal of Computer Vision. — 2020. — Т. 128, № 7. — С. 1956—1981.
6. Microsoft coco: Common objects in context [Текст] / T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick // European conference on computer vision. — Springer. 2014. — С. 740—755.
7. The cityscapes dataset for semantic urban scene understanding [Текст] / M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — С. 3213—3223.
8. Generative adversarial nets (Advances in neural information processing systems)(pp. 2672–2680) [Текст] / I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio // Red Hook, NY Curran. — 2014.
9. *Kingma, D. P.* Auto-encoding variational bayes [Текст] / D. P. Kingma, M. Welling // arXiv preprint arXiv:1312.6114. — 2013.

10. *Van Den Oord, A.* Pixel recurrent neural networks [Текст] / A. Van Den Oord, N. Kalchbrenner, K. Kavukcuoglu // International conference on machine learning. — PMLR. 2016. — С. 1747—1756.

11. *Мак-Каллок, У. С.* Логическое исчисление идей, относящихся к нервной активности [Текст] / У. С. Мак-Каллок, В. Питтс // Автоматы/Под ред. КЭ Шеннона и Дж. Маккарти.—М.: Изд-во иностр. лит. — 1956. — С. 363—384.

12. *Krizhevsky, A.* ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing 25 [Текст] / A. Krizhevsky, I. Sutskever, G. Hinton. — 2012.

13. *Sánchez, J.* High-dimensional signature compression for large-scale image classification [Текст] / J. Sánchez, F. Perronnin // CVPR 2011. — IEEE. 2011. — С. 1665—1672.

14. *Simonyan, K.* Very deep convolutional networks for large-scale image recognition [Текст] / K. Simonyan, A. Zisserman // arXiv preprint arXiv:1409.1556. — 2014.

15. Deep residual learning for image recognition [Текст] / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — С. 770—778.

16. Retinaface: Single-shot multi-level face localisation in the wild [Текст] / J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2020. — С. 5203—5212.

17. Wider face: A face detection benchmark [Текст] / S. Yang, P. Luo, C.-C. Loy, X. Tang // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — С. 5525—5533.

18. The megaface benchmark: 1 million faces for recognition at scale [Текст] / I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, E. Brossard // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — С. 4873—4882.

19. Searching for mobilenetv3 [Текст] / A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan [и др.] // Proceedings of the IEEE/CVF international conference on computer vision. — 2019. — С. 1314—1324.

20. *Tan, M.* Efficientnet: Rethinking model scaling for convolutional neural networks [Текст] / M. Tan, Q. Le // International conference on machine learning. — PMLR. 2019. — С. 6105—6114.

21. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices [Текст] / S. Chen, Y. Liu, X. Gao, Z. Han // Chinese Conference on Biometric Recognition. — Springer. 2018. — С. 428—438.

22. Mobiface: A lightweight deep learning face recognition on mobile devices [Текст] / C. N. Duong, K. G. Quach, I. Jalata, N. Le, K. Luu // 2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS). — IEEE. 2019. — С. 1—6.

23. Learning both weights and connections for efficient neural network [Текст] / S. Han, J. Pool, J. Tran, W. Dally // Advances in neural information processing systems. — 2015. — Т. 28.

24. Pruning convolutional neural networks for resource efficient inference [Текст] / P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz // arXiv preprint arXiv:1611.06440. — 2016.

25. What is the state of neural network pruning? [Текст] / D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, J. Guttag // Proceedings of machine learning and systems. — 2020. — Т. 2. — С. 129—146.

26. Distilling the knowledge in a neural network [Текст] / G. Hinton, O. Vinyals, J. Dean [и др.] // arXiv preprint arXiv:1503.02531. — 2015. — Т. 2, № 7.

27. Learning efficient object detection models with knowledge distillation [Текст] / G. Chen, W. Choi, X. Yu, T. Han, M. Chandraker // Advances in neural information processing systems. — 2017. — Т. 30.

28. *Svitov, D.* MarginDistillation: Distillation for Face Recognition Neural Networks with Margin-Based Softmax [Текст] / D. Svitov, S. Alyamkin // International Journal of Computer and Information Engineering. — 2021. — Т. 15, № 3. — С. 206—210.

29. Hua, Xs Quantization networks [Текст] / J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA. — 2019. — С. 16—20.

30. A survey of quantization methods for efficient neural network inference [Текст] / A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, K. Keutzer // arXiv preprint arXiv:2103.13630. — 2021.

31. Trainable Thresholds for Neural Network Quantization [Текст] / A. Goncharenko, A. Denisov, S. Alyamkin, E. Terentev // International Work-Conference on Artificial Neural Networks. — Springer. 2019. — С. 302—312.

32. *Zoph, B.* Neural architecture search with reinforcement learning [Текст] / B. Zoph, Q. V. Le // arXiv preprint arXiv:1611.01578. — 2016.

33. *Liu, H.* Darts: Differentiable architecture search [Текст] / H. Liu, K. Simonyan, Y. Yang // arXiv preprint arXiv:1806.09055. — 2018.

34. Learning transferable architectures for scalable image recognition [Текст] / B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — С. 8697—8710.

35. Low-power computer vision: Status, challenges, and opportunities [Текст] / S. Alyamkin, ..., D. Svitov, G. K. Thiruvathukal, B. Zhang, J. Zhang, X. Zhang, S. Zhuo [и др.] // IEEE Journal on Emerging and Selected Topics in Circuits and Systems. — 2019. — Т. 9, № 2. — С. 411—421.

36. NTIRE 2021 challenge on image deblurring [Текст] / S. Nah, ..., D. Svitov, D. Pakulich, J. Kim, J. Jeong // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2021. — С. 149—165.

37. *Свитов, Д. В.* ОПТИМИЗАЦИЯ НЕЙРОСЕТЕВОГО ДЕТЕКТОРА ДВИЖУЩИХСЯ ОБЪЕКТОВ [Текст] / Д. В. Свитов, С. А. Алямкин // Автометрия. — 2021. — Т. 57, № 1. — С. 21—30.

38. *Свитов, Д. В.* Дистилляция моделей для распознавания лиц, обученных с применением функции Софтмакс с отступами [Текст] / Д. В. Свитов, С. А. Алямкин // Автоматика и телемеханика. — 2022. — № 10. — С. 35—46.

39. *LeCun, Y.* Optimal brain damage [Текст] / Y. LeCun, J. Denker, S. Solla // Advances in neural information processing systems. — 1989. — Т. 2.

40. *Frankle, J.* The lottery ticket hypothesis: Finding sparse, trainable neural networks [Текст] / J. Frankle, M. Carbin // arXiv preprint arXiv:1803.03635. — 2018.

41. Stabilizing the lottery ticket hypothesis [Текст] / J. Frankle, G. K. Dziugaite, D. M. Roy, M. Carbin // arXiv preprint arXiv:1903.01611. — 2019.

42. Pruning neural networks without any data by iteratively conserving synaptic flow [Текст] / H. Tanaka, D. Kunin, D. L. Yamins, S. Ganguli // Advances in Neural Information Processing Systems. — 2020. — Т. 33. — С. 6377—6389.

43. Tensorflow: Large-scale machine learning on heterogeneous distributed systems [Текст] / M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin [и др.] // arXiv preprint arXiv:1603.04467. — 2016.

44. Pytorch: An imperative style, high-performance deep learning library [Текст] / A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga [и др.] // Advances in neural information processing systems. — 2019. — Т. 32.

45. Efficient Knowledge Distillation from an Ensemble of Teachers. [Текст] / T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, B. Ramabhadran // Interspeech. — 2017. — С. 3697—3701.

46. *Ba, J.* Do deep nets really need to be deep? [Текст] / J. Ba, R. Caruana // Advances in neural information processing systems. — 2014. — Т. 27.

47. *Sau, B. B.* Deep model compression: Distilling knowledge from noisy teachers [Текст] / B. B. Sau, V. N. Balasubramanian // arXiv preprint arXiv:1610.09650. — 2016.

48. Born again neural networks [Текст] / T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, A. Anandkumar // International Conference on Machine Learning. — PMLR. 2018. — С. 1607—1616.

49. Fitnets: Hints for thin deep nets [Текст] / A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, Y. Bengio // arXiv preprint arXiv:1412.6550. — 2014.

50. Learning student networks via feature embedding [Текст] / H. Chen, Y. Wang, C. Xu, C. Xu, D. Tao // IEEE Transactions on Neural Networks and Learning Systems. — 2020. — Т. 32, № 1. — С. 25—35.

51. Relational knowledge distillation [Текст] / W. Park, D. Kim, Y. Lu, M. Cho // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — С. 3967—3976.

52. Cosface: Large margin cosine loss for deep face recognition [Текст] / H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — С. 5265—5274.

53. Sphereface: Deep hypersphere embedding for face recognition [Текст] / W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — С. 212—220.

54. Arcface: Additive angular margin loss for deep face recognition [Текст] / J. Deng, J. Guo, N. Xue, S. Zafeiriou // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2019. — С. 4690—4699.

55. Triplet distillation for deep face recognition [Текст] / Y. Feng, H. Wang, H. R. Hu, L. Yu, W. Wang, S. Wang // 2020 IEEE International Conference on Image Processing (ICIP). — IEEE. 2020. — С. 808—812.

56. Shrinkteanet: Million-scale lightweight face recognition via shrinking teacher-student networks [Текст] / C. N. Duong, K. Luu, K. G. Quach, N. Le // arXiv preprint arXiv:1905.10620. — 2019.

57. *Nekhaev, D.* Margin based knowledge distillation for mobile face recognition [Текст] / D. Nekhaev, S. Milyaev, I. Laptev // Twelfth International Conference on Machine Vision (ICMV 2019). Т. 11433. — SPIE. 2020. — С. 172—179.

58. Quantization and training of neural networks for efficient integer-arithmetic-only inference [Текст] / B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — С. 2704—2713.

59. Fast adjustable threshold for uniform neural network quantization [Текст] / A. Goncharenko, A. Denisov, S. Alyamkin, E. Terentev // International Journal of Computer and Information Engineering. — 2019. — Т. 13, № 9. — С. 491—495.

60. *McCulloch, W. S.* A logical calculus of the ideas immanent in nervous activity [Текст] / W. S. McCulloch, W. Pitts // The bulletin of mathematical biophysics. — 1943. — Т. 5. — С. 115—133.

61. *Riemann, B.* Ueber die Darstellbarkeit einer Function durch eine trigonometrische Reihe [Текст] / B. Riemann. — Dieterichschen Buchhandlung, 1867.

62. *Bengio, Y.* Estimating or propagating gradients through stochastic neurons for conditional computation [Текст] / Y. Bengio, N. Léonard, A. Courville // arXiv preprint arXiv:1308.3432. — 2013.

63. Mobilenets: Efficient convolutional neural networks for mobile vision applications [Текст] / A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam // arXiv preprint arXiv:1704.04861. — 2017.

64. Ssd: Single shot multibox detector [Текст] / W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg // Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. — Springer. 2016. — С. 21—37.

65. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation [Текст] / X. Chen, L. Xie, J. Wu, Q. Tian // Proceedings of the IEEE/CVF international conference on computer vision. — 2019. — С. 1294—1303.

66. Hierarchical representations for efficient architecture search [Текст] / H. Liu, K. Simonyan, O. Vinyals, C. Fernando, K. Kavukcuoglu // arXiv preprint arXiv:1711.00436. — 2017.

67. Progressive neural architecture search [Текст] / C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, K. Murphy // Proceedings of the European conference on computer vision (ECCV). — 2018. — С. 19—34.

68. Mnasnet: Platform-aware neural architecture search for mobile [Текст] / M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q. V. Le // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2019. — С. 2820—2828.

69. *Cai, H.* Proxylessnas: Direct neural architecture search on target task and hardware [Текст] / H. Cai, L. Zhu, S. Han // arXiv preprint arXiv:1812.00332. — 2018.

70. *Chen, T.* Net2net: Accelerating learning via knowledge transfer [Текст] / T. Chen, I. Goodfellow, J. Shlens // arXiv preprint arXiv:1511.05641. — 2015.

71. Network morphism [Текст] / T. Wei, C. Wang, Y. Rui, C. W. Chen // International conference on machine learning. — PMLR. 2016. — С. 564—572.

72. Practical block-wise neural network architecture generation [Текст] / Z. Zhong, J. Yan, W. Wu, J. Shao, C.-L. Liu // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — С. 2423—2432.

73. Rich feature hierarchies for accurate object detection and semantic segmentation [Текст] / R. Girshick, J. Donahue, T. Darrell, J. Malik // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2014. — С. 580—587.

74. *Girshick, R.* Fast r-cnn [Текст] / R. Girshick // Proceedings of the IEEE international conference on computer vision. — 2015. — С. 1440—1448.

75. Faster r-cnn: Towards real-time object detection with region proposal networks [Текст] / S. Ren, K. He, R. Girshick, J. Sun // Advances in neural information processing systems. — 2015. — Т. 28.

76. You only look once: Unified, real-time object detection [Текст] / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — С. 779—788.

77. *Redmon, J.* YOLO9000: better, faster, stronger [Текст] / J. Redmon, A. Farhadi // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — С. 7263—7271.

78. *Redmon, J.* Yolov3: An incremental improvement [Текст] / J. Redmon, A. Farhadi // arXiv preprint arXiv:1804.02767. — 2018.

79. *Stauffer, C.* Adaptive background mixture models for real-time tracking [Текст] / C. Stauffer, W. E. L. Grimson // Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149). Т. 2. — IEEE. 1999. — С. 246—252.

80. *St-Charles, P.-L.* SuBSENSE: A universal change detection method with local adaptive sensitivity [Текст] / P.-L. St-Charles, G.-A. Bilodeau, R. Bergevin // IEEE Transactions on Image Processing. — 2014. — Т. 24, № 1. — С. 359—373.

81. *Babaee, M.* A deep convolutional neural network for video sequence background subtraction [Текст] / M. Babaee, D. T. Dinh, G. Rigoll // Pattern Recognition. — 2018. — Т. 76. — С. 635—649.

82. *Lim, L. A.* Learning multi-scale features for foreground segmentation [Текст] / L. A. Lim, H. Y. Keles // Pattern Analysis and Applications. — 2020. — Т. 23, № 3. — С. 1369—1380.

83. *Johnson, J.* Perceptual losses for real-time style transfer and super-resolution [Текст] / J. Johnson, A. Alahi, L. Fei-Fei // Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. — Springer. 2016. — С. 694—711.

84. Unsupervised real-world image super resolution via domain-distance aware training [Текст] / Y. Wei, S. Gu, Y. Li, R. Timofte, L. Jin, H. Song // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2021. — С. 13385—13394.

85. *Fritsche, M.* Frequency separation for real-world super-resolution [Текст] / M. Fritsche, S. Gu, R. Timofte // 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). — IEEE. 2019. — С. 3599—3608.

86. CDnet 2014: An expanded change detection benchmark dataset [Текст] / Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar // Proceedings of the IEEE conference on computer vision and pattern recognition workshops. — 2014. — С. 387—394.

87. *Braham, M.* Deep background subtraction with scene-specific convolutional neural networks [Текст] / M. Braham, M. Van Droogenbroeck // 2016 international conference on systems, signals and image processing (IWSSIP). — IEEE. 2016. — С. 1—4.

88. Static and moving object detection using flux tensor with split Gaussian models [Текст] / R. Wang, F. Bunyak, G. Seetharaman, K. Palaniappan // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. — 2014. — С. 414—418.

89. *Lim, L. A.* Foreground segmentation using convolutional neural networks for multiscale feature encoding [Текст] / L. A. Lim, H. Y. Keles // Pattern Recognition Letters. — 2018. — Т. 112. — С. 256—262.

90. Noscope: optimizing neural network queries over video at scale [Текст] / D. Kang, J. Emmons, F. Abuzaid, P. Bailis, M. Zaharia // arXiv preprint arXiv:1703.02529. — 2017.

91. *Yu, R.* Remotenet: Efficient relevant motion event detection for large-scale home surveillance videos [Текст] / R. Yu, H. Wang, L. S. Davis // 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). — IEEE. 2018. — С. 1642—1651.

92. Mobilenetv2: Inverted residuals and linear bottlenecks [Текст] / M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — С. 4510—4520.

93. Going deeper with convolutions [Текст] / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2015. — С. 1—9.

94. URL: https://github.com/david-svitov/AmphibianDetector. — (Дата обр. 15.01.2023).

95. *Bridle, J.* Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters [Текст] / J. Bridle // Advances in neural information processing systems. — 1989. — Т. 2.

96. *Грабовой, А. В.* Байесовская дистилляция моделей глубокого обучения [Текст] / А. В. Грабовой, В. В. Стрижов // Автоматика и телемеханика. — 2021. — № 11. — С. 16—29.

97. *Грабовой, А. В.* Вероятностная интерпретация задачи дистилляции [Текст] / А. В. Грабовой, В. В. Стрижов // Автоматика и телемеханика. — 2022. — № 1. — С. 150—168.

98. *Huang, Z.* Like what you like: Knowledge distill via neuron selectivity transfer [Текст] / Z. Huang, N. Wang // arXiv preprint arXiv:1707.01219. — 2017.

99. Joint face detection and alignment using multitask cascaded convolutional networks [Текст] / K. Zhang, Z. Zhang, Z. Li, Y. Qiao // IEEE signal processing letters. — 2016. — Т. 23, № 10. — С. 1499—1503.

100. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition [Текст] / Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao // Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. — Springer. 2016. — С. 87—102.

101. *Robbins, H.* A stochastic approximation method [Текст] / H. Robbins, S. Monro // The annals of mathematical statistics. — 1951. — С. 400—407.

102. URL: https://github.com/david-svitov/margindistillation. — (Дата обр. 15.01.2023).

103. *Ng, H.-W.* A data-driven approach to cleaning large face datasets [Текст] / H.-W. Ng, S. Winkler // 2014 IEEE international conference on image processing (ICIP). — IEEE. 2014. — С. 343—347.

104. URL: https://pages.nist.gov/frvt/html/frvt11.html. — (Дата обр. 15.01.2023).

105. *Ramachandran, P.* Searching for activation functions [Текст] / P. Ramachandran, B. Zoph, Q. V. Le // arXiv preprint arXiv:1710.05941. — 2017.

106. *Hu, J.* Squeeze-and-excitation networks [Текст] / J. Hu, L. Shen, G. Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — С. 7132—7141.

Список рисунков

Список таблиц

УТВЕРЖДАЮ
Технический директор
ООО “Экспасофт”, к.т.н
Алямкин Сергей Анатольевич
«24» января 2022 г.

АКТ о внедрении результатов диссертационной работы Свитова Давида Вячеславовича в технологию распознавания лиц ООО “Экспасофт”

Настоящий акт составлен о том, что результаты диссертационного исследования аспиранта “Института автоматики и электрометрии” Свитова Давида Вячеславовича на тему “Оптимизация производительности свёрточных нейронных сетей без потери точности” используются в технологии распознавания лиц ООО “Экспасофт”, внедрённой такими компаниями как:

- ООО “Новотелеком”
- ООО “Рубитек РУС”
- ООО “Открытая мобильная платформа”

Технический директор
ООО ”Экспасофт”
Алямкин С. А

Общество с ограниченной ответственностью "Экспасофт" РОССИЯ г.Новосибирск