Федеральное государственное автономное образовательное учреждение
высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

На правах рукописи

АХМЕТЗЯНОВ КИРИЛЛ РАИСОВИЧ

НЕЙРО-СЕТЕВЫЕ МЕТОДЫ И АЛГОРИТМЫ САМООБУЧЕНИЯ ПРИ ОБРАБОТКЕ ДАННЫХ В СИСТЕМЕ АВТОМАТИЗАЦИИ ПРОЦЕССА СОРТИРОВКИ БЫТОВЫХ ОТХОДОВ

05.13.06 – Автоматизация и управление

технологическими процессами и производствами (в промышленности)

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:

доктор технических наук, профессор

Южаков Александр Анатольевич

Пермь 2021

Оглавление

Введение

Актуальность темы исследования и степень ее разработанности.

В России ежегодно в среднем накапливается 3 млн.т пластиковых отходов, и это количество с годами увеличивается, а время разложения пластика может достигать 1000 лет. В то же время количество перерабатываемого пластика в стране составляет менее 7%. Существующие заводы по переработке пластика не справляются с огромным потоком мусора. Для переработки пластика необходимо отсортировать его по видам пластиковых отходов, например, пластиковые бутылки, так как у каждого пластика своя температура плавления. Поэтому важно в режиме «реального времени» как можно точнее определять эти предметы среди прочего мусора, которые состоят из пластика.

В настоящее время этап сортировки пластиковых материалов на предприятиях по переработке отходов является одним из наименее автоматизированных этапов технологической цепочки. Для увеличения пропускной способности подобных предприятий необходимо разработать аппаратурно-программные средства по сортировке пластиковых материалов.

В информационных технологиях определение предметов называется классификацией объектов. Классификация объектов — это алгоритмы и набор математических преобразований, которые позволяют определить (идентифицировать) предмет.

Значительный вклад в развитие «умных» городов и автоматизации процесса сортировки бытовых отходов внесли такие зарубежные ученые, как A.R. Al-Ali, H. Basri, D.A. Wahab, H. Fu и W. Yousef, так и отечественные ученные – М.В. Соколов, Б.Б. Бобович, О.Б. Ганин, С.В. Абламейко, А.П. Добрынин, С.Н. Максимов и др.

Ученые T. Raiko и N. Krishnan применили и адаптировали технологии машинного и глубокого обучения, а также компьютерного зрения в области сортировки отходов. T. Raiko является разработчиком автоматизированной системы сортировки отходов. Его технология заключается в фотографировании мусора с помощью видеокамеры, применении нейронных сетей и компьютерного зрения для сортировки отходов по категориям и материалу.

Существует множество методов компьютерного зрения для распознавания объектов. К классическим методам относятся SIFT, SURF, LBP и HOG. Эти методы обладают следующими преимуществами: достаточно высокое быстродействие, устойчивость к контрастности изображения, поворотам, масштабам и частичному закрытию объектов распознавания. К недостаткам относится то, что объекты без ярко выраженной текстуры и с фрактальной структурой будут неверно распознаваться.

Другим классом методов распознавания являются методы на основе сверточных нейронных сетей (наиболее известные LeNet, AlexNet, VGG, ResNet, MobileNet). Преимуществом этих методов является возможность аппроксимировать любую функцию и обучить модель классифицировать любые объекты. К недостаткам относится то, что необходимо проводить вычислительно затратную процедуру поиска оптимальных гиперпараметров, которые для каждого набора обучающей выборки могут различаться.

В диссертационной работе исследуется задача распознавания объектов на основе изображений в условиях ограниченных вычислительных ресурсов при выборе оптимальных гиперпараметров. Для ее реализации предлагается использовать многозадачный многокритериальный метод гиперпараметрической оптимизации. Такой метод позволит выбрать гиперпараметры обучения математических моделей распознавания на основе нескольких заданных критериев (достигаемая максимальная точность распознавания и время обучения модели) и нескольких задач распознавания. Это дает возможность применения сверточных нейронных сетей при

меньших требованиях к вычислительной платформе без снижения качества распознавания.

Объект исследования: автоматизированный процесс предварительной сортировки бытовых отходов для переработки на предприятиях отраслей промышленности.

Предмет исследования: научно-методический аппарат автоматической классификации изображений с помощью сверточных нейронных сетей.

Цель работы: совершенствование автоматизированной сортировки бытовых отходов по заданным критериям качества на основе разработки и внедрения методов и алгоритмов самообучения систем автоматизации процессов сортировки с применением сверточных нейронных сетей.

Задачи работы. Для достижения поставленной цели необходимо решить следующие задачи:

1) провести анализ принципов функционирования существующих систем автоматизированной сортировки бытовых отходов;

2) построить критерии оптимизации гиперпараметров для автоматизированной сортировки бытовых отходов;

3) разработать метод оптимизации сверточной нейронной сети с комплексом заданных гиперпараметров для повышения качества процесса обучения;

4) разработать метод оптимизации вычислительных затрат при построении модели автоматической классификации изображений бытовых отходов;

5) разработать метод автоматического обучения распознавания при сортировки бытовых отходов;

6) внедрить предложенные методы оптимизации гиперпараметров и вычислительных затрат в устройство по автоматизированной сортировке бытовых отходов.

Методы исследования основаны на математических и методических принципах построения нейронных сетей, теории машинного обучения, обработки изображений, распознавания образов, теории оптимизации, теории планирования и обработки экспериментальных данных.

Основные положения, выносимые на защиту:

1. Предложенные критерии для оптимизации гиперпараметров моделей классификации изображений (п. 15. Теоретические основы, методы и алгоритмы интеллектуализации решения прикладных задач при построении АСУ широкого назначения (АСУТП, АСУП, АСТПП и др.).

2. Метод многозадачной многокритериальной гиперпараметрической оптимизации вычислений математической модели (п. 15. Теоретические основы, методы и алгоритмы интеллектуализации решения прикладных задач при построении АСУ широкого назначения (АСУТП, АСУП, АСТПП и др.)).

3. Метод оптимизации вычислительных затрат при построении модели автоматической классификации изображений бытовых отходов (п. 15. Теоретические основы, методы и алгоритмы интеллектуализации решения прикладных задач при построении АСУ широкого назначения (АСУТП, АСУП, АСТПП и др.)).

4. Метод автоматического обучения специализированной нейронной сети для сортировки мусора (п. 8. Формализованные методы анализа, синтеза, исследования и оптимизация модульных структур систем сбора и обработки данных в АСУТП, АСУП, АСТПП и др.)

5. Программный комплекс с учетом разработанных методов в составе устройства по автоматизированной сортировке бытовых отходов (п. 15. Теоретические основы, методы и алгоритмы интеллектуализации решения прикладных задач при построении АСУ широкого назначения (АСУТП, АСУП, АСТПП и др.)).

Научная новизна:

1. Состав критериев для многокритериальной оптимизации при обучении моделей классификации изображений, отличающийся тем, что он сформирован с учетом гиперпараметров обучения и позволяет повысить точность модели классификации и снизить затраты вычислительных ресурсов автоматизированной системы.

2. Новый метод многокритериальной оптимизации гиперпараметров нейронной сети, отличающийся многозадачностью, что позволяет получить оптимальные гиперпараметры для заданных с их учетом критериев.

3. Новый метод оптимизации вычислений на основе квантования, отличающийся заданием нескольких критериев оптимизации, что позволяет уменьшить размер модели классификации без потери ее точности.

4. Новый метод сортировки бытовых отходов, отличающийся автоматическим самообучением предложенной нейронной сети, что позволяет снизить вычислительные затраты при оперативном обучении сети на примерах (данных), представляемых организациями, производящие сортировку отходов.

5. Программный комплекс в составе устройства по автоматизированной сортировке бытовых отходов, включающий нейро-сетевые методы и алгоритмы самообучения, который не требует дополнительных капитальных вложений в оборудование системы и обеспечивает повышение качества сортировки бытовых отходов.

Теоретическая значимость:

Создание научно-методического аппарата, заключающегося в разработке критериев оптимизации гиперпараметров моделей классификации изображений, построении новых методов многозадачной многокритериальной гиперпараметрической оптимизации, оптимизации вычислений на основе квантования и автоматического обучения специализированной нейронной сети для

сортировки мусора. Разработанные методы и алгоритмы применимы в других областях, требующих автоматизации с применением методов компьютерного зрения.

Практическая значимость заключается:

1) в разработке и программной реализации автоматизированной системы по сортировке бытовых отходов на основе предложенных методов, что позволило уменьшить общее время обучения нейронной сети в условиях ограниченных вычислительных ресурсов на 10% при испытаниях (в «Сортомате»), при сохранении точности полученной нейронной сети; повысить эффективность используемых вычислительных ресурсов АСУТП для распознавания объекта на изображении на 15% при испытаниях (в «Сортомате»);

2) в возможности использования затратных эффективных алгоритмов распознавания объектов в промышленных АСУТП с ресурсными ограничениями на предприятиях по переработке бытовых отходов без потери точности распознавания;

3) в применимости разработанных методов и алгоритмов в подсистемах компьютерного зрения автоматизации производственных процессов без изменений разработанных методов и алгоритмов.

Достоверность и обоснованность результатов. Общие тенденции, полученные в результате исследования, не противоречат результатам, представленных в литературе другими исследователями, а также подтверждаются сопоставлением теоретических выводов с результатами имитационных экспериментов и результатами внедрения устройства по автоматизированной сортировке бытовых отходов «Сортомат».

Апробация работы. Основные результаты диссертационной работы представлялись и обсуждались на XXI международной конференции по мягким вычислениям и измерениям (Россия, г. Санкт-Петербург, 2018 г.), IV и V всероссийской научно-практической конференции «Искусственный интеллект в решении актуальных социальных и экономических проблем XXI века» (Россия, г.

Пермь, 2019 и 2020 г.), 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (Россия, г. Санкт-Петербург, 2020 г.), 2019 и 2020 International Russian Automation Conference (Россия, г. Сочи, 2019 и 2020 г.).

Работы по теме диссертационного исследования выполнялись в рамках научного проекта № С-26/174.6 международной исследовательской группы учёных (МИГ-30).

Публикации. Основные результаты диссертационной работы опубликованы в 14 научных работах, из них: 7 статей в журналах, входящих в перечень ведущих журналов и изданий, рекомендуемых ВАК; 3 в изданиях, индексируемых в базах SCOPUS; 1 патент на полезную модель и 1 свидетельство о регистрации программы для ЭВМ, остальные – в тезисах докладов, материалах конференций и прочих источниках.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы из 127 наименований и 8 приложений. Полный объем диссертации составляет 121 страница, из которых 96 страниц занимает основной текст диссертации, включающий 21 рисунок и 15 таблиц.

Глава 1. Анализ объекта исследования и постановка задачи

Переработка отходов может быть рассмотрена в двух контекстах: технологический и социально-экономический. Современная промышленность и современные технологии позволяют организовать процесс переработки отходов более эффективно как с точки зрения времени, так и с точки зрения денег. С другой стороны, у современного поколения есть потребность в балансе между жизнью и работой. Современные технологии позволяют решить потребности общества.

1.1 Современной состояние промышленности

Развитие промышленности имеет долгую историю. С появлением новых научных открытий и развитием научно-технического прогресса промышленность становилась более эффективной, а человеческий труд видоизменялся. В 17 веке произошел переход от ручного труда на использование паровых машин. С появлением электричества в 19 веке произошел переход на массовое производство и использование конвейеров. Затем появились вычислительные машины и процессоры, и в 70-е годы 20 века произошел переход на автоматизированное управление производством. В настоящее время происходит переход на новый способ организации производства.

Последний переход также называют Индустрией 4.0 [10,11]. Это концепция, предложенная правительством Германии в 2011 году [12,13]. Индустрия 4.0 позволяет быстрее масштабировать предприятия за счет получения аналитики с разных предприятий и применению машинного обучения для улучшения развития как последующих предприятий, так и предыдущих. Накопление большого количество информации с различных предприятий позволяет, с одной стороны, сделать

предприятие универсальным, а с другой - подстраиваться под специфические условия. Индустрия 4.0 объединят потребителей, логистику и производителей в единую систему, что дает дополнительные возможности для планирования закупок, производства и лучшее понимание потребностей потребителей, что, в свою очередь, повышает производительность производства различной продукции на всех ее этапах и уменьшает издержки и простои производства.

Появление нового вида индустрии означает появление и нового способа организации работы. Работа 4.0 [12,13] обозначает интегрированный труд, растущая взаимосвязь и рост сотрудничества между человеком и машиной, сдвиг в ценностях и новые социальные компромиссы. Работа 4.0 дает возможность рабочему меньше проводить время на работе и работать удаленно, из дома, и проводить больше времени с семьей. Работа становится удаленной, не является приоритетной, а является лишь частью жизни. Работа становится более безопасной, а также требует переквалификации. Работа является менее рутинной и более интеллектуальной, зарплата более высокой. Инженер обслуживает не одно предприятие, а одновременно несколько. Появление новой формы организации труда является логичным продолжением научно-технического прогресса и потребностей современного поколения.

1.2 Современная переработка отходов

Развитие промышленности отражается и на переработке отходов. В переработке отходов можно выделить несколько этапов развития.

Нулевой этап – складирование и сжигание отходов. Первый этап – ручная сортировка смешанного мусора. Второй этап – автоматизированная сортировка, при котором предприятия существуют отдельно друг от друга. Третий этап – объединение

предприятий в сеть, сортировка осуществляется с помощью роботов-манипуляторов, с различными датчиками, человек участвует в разметке данных, проверяет правильность сортировки. Четвертый этап – автоматизированная сортировка, при котором предприятия по переработке отходов обучают друг друга и не требуется ручного вмешательства в процесс проверки правильности работы сортировщика. Третий и четвертый этап является Индустрией 4.0. На рисунке 1.1 представлена схема взаимодействия между предприятиями третьего и четвертого этапа развития.

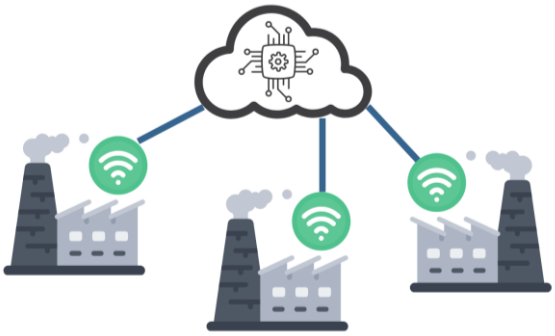

Рисунок 1.1 – «Умные» предприятия взаимосвязаны через центральный сервер

Складирование отходов означает вывоз мусора за пределы города на полигоны без их дальнейшей переработки и извлечения экономической выгоды. Отходы на полигонах могут причинить вред экологии, так как продукты разложения загрязняют почву и грунтовые воды. Сжигание отходов означает уничтожение в печах. При сжигании мусора продукты горения также вредят экологии. Существуют технологии, при которых на выходе процесса сжигания получается только тепло, пар и зола. Тепло можно использовать для отопления жилых помещений либо преобразовывать в электроэнергию. В этом случае отходы перерабатываются в энергию, что носит экономический эффект и приносит дополнительную прибыль предприятиям по переработке отходов [14].

При первом этапе развития переработки отходов используется ручной труд. Труд и навыки рабочего применяются для сортировки мусора по различным

категориям. Такой вид переработки малоэффективен с точки зрения производительности труда, а также может причинить вред здоровью рабочему из-за взаимодействия с опасными и пыльными отходами.

При втором этапе сортировку производят автоматически и такой вид сортировки более производителен, чем при ручной сортировке. К недостаткам можно отнести то, что в случае возникновения неисправности на предприятии инженеру необходимо лично присутствовать на этом предприятии, отлаживать и исправлять возникшие неисправности.

В третьем этапе предприятия по переработке отходов объединяются в единую сеть, что облегчает мониторинг и устранение неисправностей. На таких предприятиях используются интеллектуальные системы и знания, извлеченные с одного предприятия, которые могут быть перенесены на другое предприятие, что увеличивает качество сортировки. Это качество со временем работы предприятий увеличивается, так как система сортировки дообучается, использую продолжительное обучение (continual learning) [15–17]. Подобные работающие компании уже есть, например, ZenRobotics [18,19], Machinex [20] и AMP Robotics [21]. Недостатком является то, что в процессе работы такого предприятия создается огромное количество данных, которое необходимо вручную просматривать, выполнять их разметку и проверку на правильность сортировки.

Четвертый этап переработки отходов подразумевает отсутствие человека для разметки полученных с заводов данных. Сами предприятия обучают друг друга, используя федеративное обучение (federated learning)[22–23]. При таком обучении потребуется большое количество заводов по переработке отходов, каждый из которых имеет свой опыт работы с мусором и этот опыт передается между предприятиями, тем самым уменьшая среднюю ошибку распознавания мусора.

1.3 Управление отходами

Переработка отходов является частью более общей проблемы - управление отходами [25,26]. Управление отходами состоит из нескольких компонентов: сбор отходов, перевозка, переработка и получение переработанных отходов. Управление отходами представлено на рисунке 1.2.

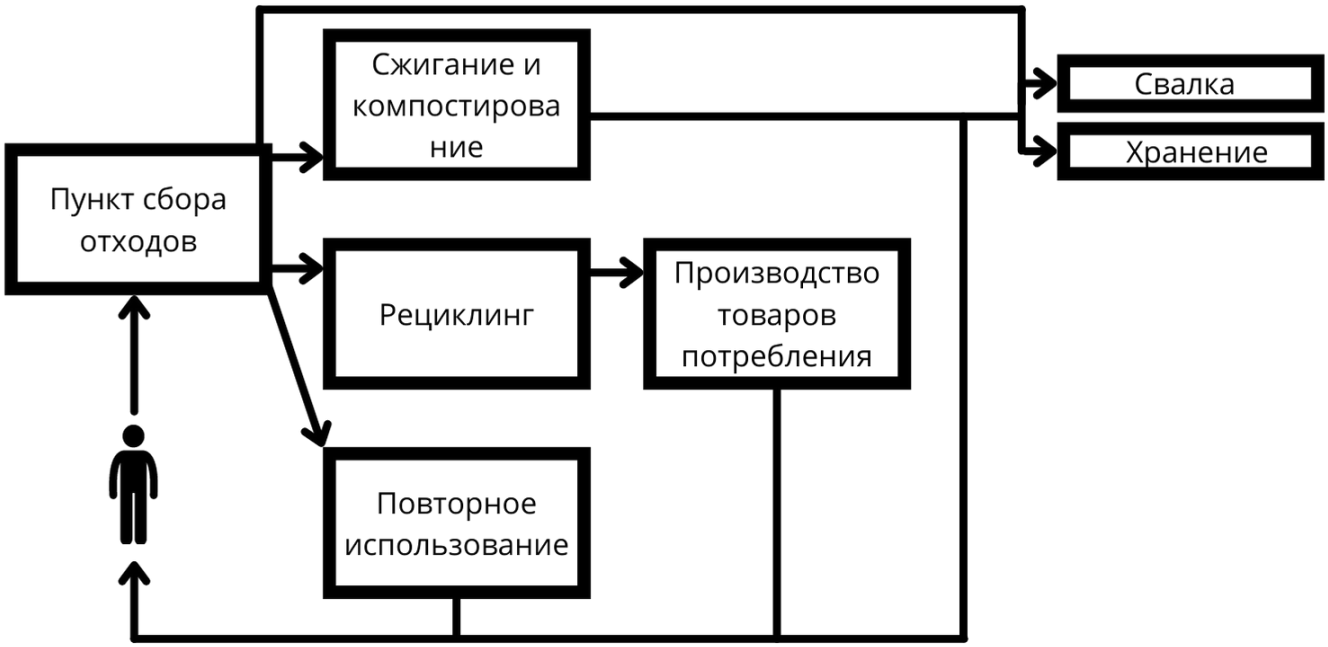

Рисунок 1.2 – Управление отходами

Схема на рисунке 1.2 состоит из потребителя и производителя мусора (обозначен человечком), пунктов сбора отходов, сжигающего и компостирующего предприятия, предприятия, проводящего рециклинг (восстановление материалов), производителей товаров потребления, компаний, выполняющие подготовку и перепродажу использованных вещей, свалки и хранилища.

С точки зрения экологии наиболее благоприятным для окружающей среды является использование вещей повторно [27], когда некоторые ненужные одному

человеку вещи (например, одежда, бытовая техника) может быть нужна другому человеку. Компании могут производить сбор таких вещей, производить их подготовку (очистку, проверку состояния вещей) и продавать по относительно низкой цене.

Другой вариант обращения с отходами – рециклинг [27–29], который заключается в предварительной сортировке, физический и термической обработке отходов с получением промежуточного продукта (например, при рециклинге пластика получается флекс и гранулят). Затем этот продукт продается компаниям, которые в свою очередь производят товары потребления (например, для пластика это могут бутылки) и продают их потребителям.

Еще одним вариантов переработки отходов является сжигание [30] и компостирование [31-33]. Такие предприятия подвергают отходу высоким температурам и на выходе получают электроэнергию, тепло и газ, которые продаются пользователям. Также выходом является неснижаемый остаток - зола, часть которой можно отправить на изготовление различных товаров (например, асфальта), другая часть (тяжелые металлы и токсины) отправляются на свалки или на хранение с целью переработки в будущем при появлении необходимой технологии.

Последним вариантом обращение с отходами является прямое отправление отходов на свалки [34,35] и хранилища отходов [36].

Сбор отходов подразумевает работу с населением [37,38]. Раздельный сбор мусора (например, на органический и неорганический) более выгоден для предприятий, так как устраняет необходимость в предварительной сортировке мусора. Для достижения этого возможна работа средств массовой информация для прививания культуры раздельного сбора мусора, а также материальной поощрения в виде различных скидок и бонусов для последующей покупки товаров в магазинах.

Перевозка мусора означает умение работать с раздельным мусором и оптимизацию логистики. Экономически более выгодным является размещение

перерабатывающих предприятий как можно ближе к городу для оптимизации логистики [39,40], а также перевозка только заполненных мусорных ящиков.

Переработка мусора означает преобразование мусора в некий переработанный материал (например, для пластика - гранулят), который в дальнейшем продается другим предприятиям, которые в свою очередь создают продукцию.

Таким образом, при грамотном проектировании процесса управления отходами получается социальная выгода (для населения - своевременный вывоз мусора, для рабочих - работа, не отнимающая много времени для более активной жизни и соблюдение баланса жизнь-работа) и экономическая выгода (для населения - материальная выгода от раздельного сбора отходов, для предпринимателей - экономия на переработке отходов и продажа новой произведенной продукции).

1.4 Исследование и анализ технологического процесса сортировки мусора

Рассмотрим современное предприятия по переработке отходов более детально. Оно состоит из нескольких компонентов: конвейерная лента, сенсоры, математическая модель, компьютер, контроллер, робот и роутер. На рисунке 1.3 представлена схема работы автоматизированной сортировки современного предприятия по переработке отходов.

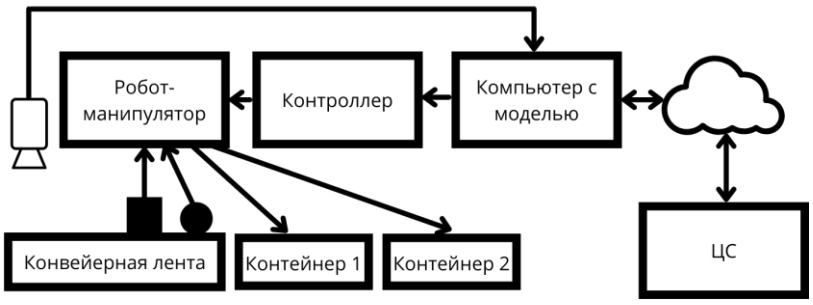

Рисунок 1.3 – Схема работы автоматизированной сортировки современного предприятия по переработке отходов

Схема состоит из сенсора (обозначен в виде камеры слева посередине), робота-манипулятора, контроллера, компьютера с математической моделью, сети Интернет (обозначена в виде облака), конвейерной ленты, на которой перемещается мусор, контейнеров и центрального сервера (обозначен как ЦС). Сенсоры получают информацию из окружающего мира и передают ее в компьютер. В компьютере находится математическая модель, которая преобразует входную информацию для нахождения соответствия между этой информацией и категорией мусора. Этим соответствием может быть метка класса (материал, тип пластика), расположением мусора на конвейерной ленте, размер мусора, уверенность в метке класса и т.д. Затем эта обработанная информация передается в контроллер, который управляет роботом-манипулятором для выполнения механической сортировки мусора в соответствующий контейнер. Через Интернет на центральный сервер передаются данные, полученные с сенсора. На центральном сервере выполняется обновление математической модели, которая передается обратно на предприятие. Схема обновления математической модели представлено на рисунке 1.4.

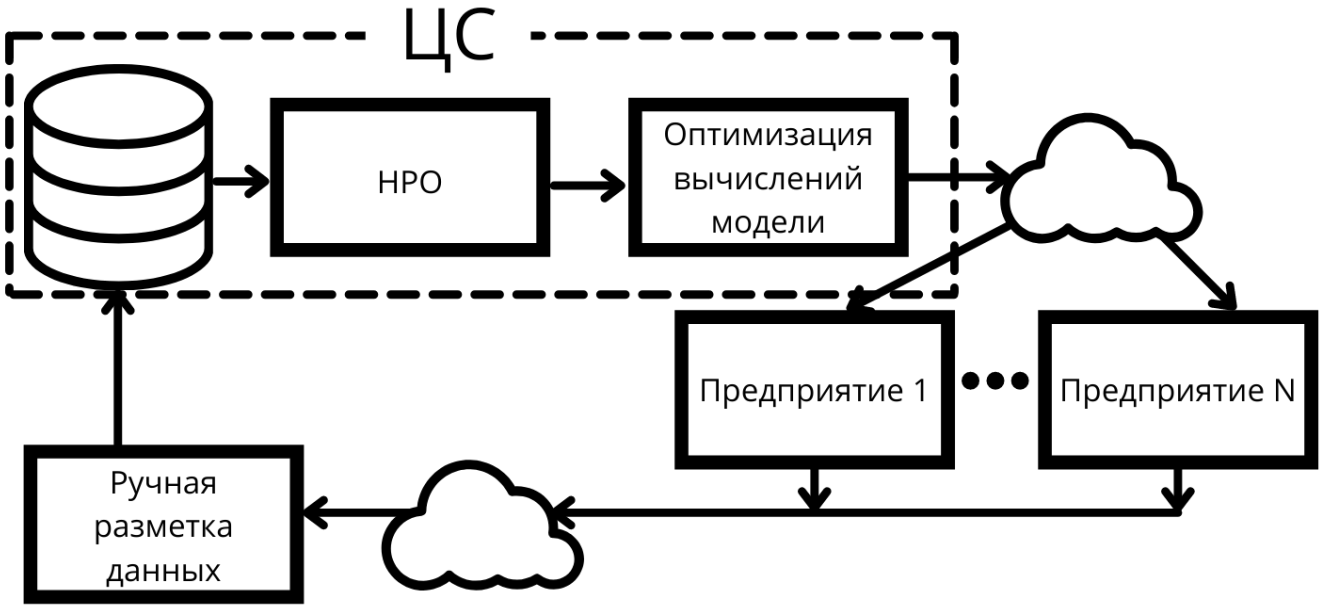

Рисунок 1.4 – Схема обновления математической модели

Схема на рисунке 1.4 состоит из центрального сервера, предприятий, сети Интернет и ручной разметки данных. В базе данных (показана в верхней левой части рисунка) хранятся размеченные данные, полученные с сенсора. Эти данные загружаются в программу, которая выполняет гиперпараметрическую оптимизацию (на рисунке обозначено как HPO) и обучение математической модели классификации данных с сенсора. Затем обученная модель оптимизируется по нескольким критериям (например, скорость и размер модели). Полученная модель передается по Интернету на компьютеры, установленные на предприятиях по переработке отходов. В процессе работы предприятий создаются новые данные, получаемые с сенсора. Эти данные передаются по Интернету и для них выполняется ручная разметка, которая может выполняться удаленно вне центрального сервера как инженерами, выполняющие обучение модели, так и нанятыми сторонними сотрудниками. Размеченные новые данные загружаются в базу данных и дополняют обучающую выборку.

Теперь необходимо ответить на следующие вопросы. Какие именно сенсоры использовать для сортировки отходов? Какую математическую модель использовать для обработки данных, полученных с сенсоров?

1.5 Выбор факторов эффективности процесса

Сортировка отходов (мусора - СМ) обладает своими особенностями, отличными от других технологических процессов производства. Устройство, производящее сортировку отходов, должно обладать высокой скоростью обработки информации при сортировке на конвейерной ленте, математический аппарат, обрабатывающий входную информацию, должен обладать устойчивостью при изменении геометрической формы классифицируемого предмета (мусора), при его загрязнении,

а также должен быть легко перенастраиваемым при появлении каких-либо новых видов отходов.

1.6 Обзор методов СМ

Существуют различные методы сортировки мусора, основанные на физических и химических свойствах материалов. Каждый датчик и сенсор имеет свои преимущества и недостатки, которые необходимо учитывать при выборе датчика или сенсора, который бы подходил для сортировки пластика и был бы также достаточно дешев.

1.6.1 Сортировка с помощью NIR-датчиков

ИК-спектрограф пропускает через образец мусора волны разной частоты, при пропускании происходит возбуждение колебательных движений молекул, при этом происходит поглощение света и для разных молекул максимальное поглощение достигается на разных частотах света. Зная, на какой частоте происходит максимальное поглощение, можно определить состав вещества [41].

Достоинства NIR-датчиков:

1. Существуют базы ИК-спектров веществ [42,43], что позволяет правильно идентифицировать большое количество веществ.

2. Фурье-ИК-спектрографы позволяют получать спектры за короткий промежуток времени (50 спектров в секунду).

Недостатки NIR-датчиков:

1. Если частота света, при которой происходит максимальное поглощение, находится вне диапазона чувствительности спектрографа, то определить вещество не удастся, так как не удастся зафиксировать частоту света максимального поглощения.

2. Наличие помех в спектре из-за воздуха и различных вкраплений других веществ в образце.

1.6.2 Электростатическая сортировка

Метод основан на электростатических свойствах веществ. Частицы веществ материала отдают свои заряды быстрее или медленнее в зависимости от проводящих свойств поверхности частиц [44].

Преимущества электростатической сортировки:

1. Возможность сортировать материалы одинаковые по плотности.

2. Низкое энергопотребление при высокой производительности.

Недостатки электростатической сортировки:

1. Метод пригоден для сухих и относительно чистых материалов [44].

2. Пригоден для пластмасс, состоящих из двух и меньшего количества компонентов.

1.6.3 Сортировка с использованием рентгеновских датчиков

Метод основан на свойствах вещества поглощать рентгеновское излучение.

Преимущества рентгеновских датчиков:

1. Позволяет точно идентифицировать ПВХ и ПЭТ (ПВХ – поливинилхлорид, ПЭТ – полиэтилентерефталат) [45].

2. Правильно идентифицирует даже при наличии этикеток и загрязнении на бутылке.

Недостатки рентгеновских датчиков:

1. Не отличает ПЭТ от ПЭ (ПЭ – полиэтилен).

2. Рентгеновское излучение вредно для здоровья, людей из-за чего необходимо персоналу, работающему с этим датчиком, применять средства защиты.

1.6.4 Сортировка с использованием датчиков в видимом диапазоне

Метод основан на улавливании света в видимом диапазоне, то есть того света, который видит человеческий глаз.

Преимущества датчиков в видимом диапазоне:

1. Низкая стоимость датчиков.

2. Простота настройки (необходимо установить датчик и настроить фокусное расстояние до объекта сортировки).

Недостатки датчиков в видимом диапазоне:

1. Предметы, которые внешне неотличимы, но имеющие разную структуры, будут неправильно сортированы.

2. Предметы, которые полностью утратили исходную форму (например, при дроблении), будут неправильно сортированы.

С учетом вышеизложенного, а также из-за низкой стоимости, универсальности и возможности со временем пополнять базу данных новыми видами отходов предлагается использовать для сортировки мусора датчик в видимом диапазоне.

1.7 Обзор методов оптической СМ

Оптические методы сортировки являются относительно простыми. Такие методы включают в себя обработку цвета и формы и позволяют определить для этой обработанной информации класс объекта для последующего принятия решения о категории предмета на изображении.

В [46] описана система сортировки по цвету пластика. В этой системе используются три светодиода (красный, зеленый, синий), фоторезистор и микроконтроллер PIC16F877. Предложена система сортировки бумага и не-бумага [47], которая является частью системы, включающая робота-манипулятора. В [48] представлена система по обнаружению дефектов пустых стеклянных бутылок. Система состоит из следующих компонентов: фотографирование бутылки сверху при освещении сверху, фото вверху при освещении снизу, фото сборку при освещении сбоку, обнаружение границ бутылок, вырезание частей фотографий для последующей обработки полносвязной нейронной сетью с одним скрытым слоем, которая делает вывод: имеет ли бутылка дефект или нет. Система сортировки твердых частиц по их форме и размеру реализована в [49]. Система состоит из конвеерной ленты, лазера и камеры. Камера фиксирует глубину просвечивания лазера, затем по полученным с камеры данным вычисляется форма частицы. В статье [50] предлагают метод классификации ПЭТ бутылок с помощью гистограммы распределения цветов. Такой метод способен распознавать в том числе скрученные и деформированные бутылки. Метод определяет прозрачность бутылки и распознает непрозрачную бутылку как не ПЭТ. В [51] описана система распознавания для автомата по приему вторсырья. Эта система использует бинокулярное зрение и иерархическую систему распознавания для уменьшения времени обработки изображений. В [52] создан оптический метод

классификации ПЭТ и тетрапак. Метод основан на гистограмме черно-белого изображения. Этот метод инвариантен к поворотам и масштабированию объекта.

1.8 Обзор нейро-сетевых технологий СМ

Нейро-сетевые технологии сортировки являются продолжение оптических методов сортировки. Такие методы автоматически обучаются с помощью метода обратного распространения ошибки и находят признаки на изображении, по которым можно отличить различные категории мусора. Нейро-сетевые методы вычислительно более сложные, но дают наибольшую точность по сравнению с «чистыми» оптические методами и не требуют для каждой задачи классификации вручную искать отличительные признаки каждой категории мусора.

В работе [53] предлагается метод для классификации Polystyrene (PS), Polycarbonate (PC), Polypropylene (PP), Acrylonitrile Butadiene Styrene (ABS) и High Impact Polystyrene Sheet (HIPS), основанный на применении гистограммы цветов и полносвязной нейронной сети с одним скрытым слоем. В [54] представлены результаты исследования по сортировке пластика, бумаги и металла с помощью методов компьютерного зрения. В качестве входа используются цветные изображения, полученные с камеры Raspberry Pi. Для обработки выбирают метод между двумя: AlexNet и k-means с SVM. AlexNet – это архитектура сверточной нейронной сети, которая опередила ранее существовавшие решения на ежегодном соревнований по распознаванию изображений ImageNet Large Scale Visual Recognition Challenge. K-means – это метод кластеризации с заданным количеством кластеров k. SVM – это метод классификации, в котором максимизируется зазор между разделяющей гиперплоскостью и объектами классов. Наибольшая точность достигается с помощью второго метода (k-means с SVM). Авторами [55] описаны

эксперименты по выбору сверточной нейронной сети для аппарата по приему вторсырья (ПЭТ-бутылок и алюминиевых банок). Наибольшая точность среди AlexNet, LeNet и SqueezeNet (SqueezeNet – это архитектура сверточной нейронной сети, в которой впервые использовался модуль Fire; модуль Fire – это модуль, который состоит из слоя сжатия и расширения, что уменьшает количество параметров и увеличивает скорость работы нейронной сети) достигается с помощью AlexNet, сопоставимые результаты достигаются с помощью LeNet (LeNet – это архитектура сверточной нейронной сети для распознавания рукописных цифр) при обучении нескольких нейронных сетей по два класса в каждой (бутылка/не бутылка, банка/не банка). В статье [56] применяют нейронную сеть InceptionV3 для классификации на биоразлагаемый и небиоразлагаемый мусор. В [57] обучили сверточную нейронную сеть VGG-16. Авторами выполнена тонкая настройка сети и для увеличения точности сети используют технологию увеличение данных. Нейронная сеть запускается на Nvidia Jetson Nano и работает от солнечной энергии для классификации бумаги, пластика, картона, металла, стекла и прочего мусора.

Сверточные нейронные сети в сочетании с оптическими методами распознавании при использовании для сортировки мусора дают более точные результаты, чем другие рассмотренные методы. Необходимо определить, какая именно архитектура дает наибольшую точность классификации бытовых отходов для заданной выборки и заданных вычислительных ограничениях.

1.9 Выводы

В последнее время из-за постоянного накопления мусора проблема раздельного сбора отходов становится всё более актуальной. Автоматическая сортировка мусора выполняется на основе информации, получаемой с сенсора, которая обрабатывается

компьютером с математической моделью и соотносит эту информацию с категорией мусора. Среди рассмотренных датчиков (NIR-датчик, электростатическая сортировка, рентгеновский датчик, датчик в видимом диапазоне) особый интерес представляет датчик в видимом диапазоне из-за универсальности и низкой стоимости, что позволяет со временем пополнять базу данных и добавлять новые категории мусора.

Анализ существующих алгоритмов для обработки информации в видимом диапазоне показал, что традиционными являются SIFT, SURF и HOG. Наибольшую популярность в последнее время набирают сверточные нейронные сети, способные извлекать сложную скрытую структурированную информацию из изображений, благодаря чему достигается высокая точность классификации изображений.

Показано, что для эффективного решения задачи распознавания в условиях ограниченного вычислительного ресурса необходимо построить требуемую архитектуру нейронной сети и выбрать алгоритм обучения.

Показано, что предпочтительно использование датчиков в видимом диапазоне с применением нейро-сетевых технологий для распознавания отходов, а эффективное решение задачи распознавания в условиях ограниченного вычислительного ресурса требует построения специализированной архитектуры нейронной сети и подбора алгоритма обучения.

Глава 2. Разработка метода автоматизированного обучения специализированной нейронной сети для сортировки мусора

В настоящее время существует несколько методов обработки изображений для классификации отходов: классические методы и методы на основе нейронных сетей. Применение сверточных нейронных сетей для решения указанных выше задач становятся всё популярнее. Это связано со следующими современными тенденциями:

- современные статьи, посвященные классификации изображений, в основном связаны с нейронными сетями,

- крупные корпорации выпускают свои инструменты для работы с нейронными сетями,

- видеокарты выпускают специально для работы нейронных сетей (ведь нейронные сети потребляют много вычислительных ресурсов),

- нейронные сети – это маркетинговый ход, который делает заведомо более успешными и модными тех ученых, которые используют нейронные сети.

Обучение с учителем, которое используется для обучения классификаторов, построенных на нейронных сетях, является универсальным и удобным способом для обучения почти любого классификатора, настроенного на определённые категории мусора. Также сверточные нейронные сети дают наибольшую точность по сравнению с классическими методами, что подтверждается результатами соревнований, посвященных обработке изображений [5]. Ниже приведем необходимость использования нейронных сетей для решения задачи СМ, для чего будет проведен ряд экспериментов с различными архитектурами нейронных сетей.

2.1 Выбор сверточной нейронной сети

Как отмечается выше, основное внимание в решение задач распознавания в СМ делается на нейронные сети как математический аппарат обработки изображений.

Нейронные сети, как математический способ преобразования изображений различных объектов, который дает высокую точность, развивался постепенно, начиная с 1940-х годов и по сей день, и у них было три волны популярности.

Первая волна приходится на 1940–1960-е годы, которая связана с кибернетикой и разработкой теории биологического обучения [58]. В 1948 году Винером была издана книга «Кибернетика», в которой были описаны концепции управления, коммуникации и обработки сигналов. В 1949 году Хеббом была издана книга «Организация поведения», в которой он описал процесс физиологического обучения нейронов.

В этот период был создан нейрон Маккаллока–Питтса, которая является одной из первых моделей функционирования мозга. Появление этого нейрона обязано 20-летней работе Мак-Каллока по изучению нервной системы, который был по образованию психиатром и нейроанатомом, а также Питтса, который чуть позже присоединился к его работе. Питтс был по образованию математиком, и они вместе Мак-Каллок и Питтс создали модель нейрона [59].

Также в 1957 году Розенблатом был реализован перцептрон, который представляет собой модель человеческого восприятия [58]. Розентблатом была реализована нейронная сеть не только с одним скрытым слоем (однослойный перцептрон), но и с несколькими (многослойный перцептрон).

Следующими значимыми работами в области нейробиологии стали работы Хьюбела и Визеля. В своих исследованиях они показали, какие нейроны зрительной коры головного мозга активируются при визуальном восприятии картинки глазом.

Прежде чем информация из глаз попадет в зрительную кору мозга, она проходит несколько этапов предварительной обработки. С помощью палочек и колбочек сетчатки глаза визуальная информация преобразуется в электрические сигналы, которые поступают в ядро зрительного нерва, расположенного в определенной области промежуточного мозга. Далее сигнал поступает в первичный стволовой анализатор (верхние холмики четверохолмия). Затем информация отправляется в специализированное ядро таламуса (латеральное коленчатое тело). И только после этого сигнал поступает в зрительную кору головного мозга. Зрительная кора делится на несколько участков: V1 (первичная зрительная кора), V2 (вторичная зрительная кора) и т.д. до V7. Каждый участок зрительной коры имеет определенные функциональные назначения. V1 выделяет локальные признаки изображения. V2 начинает группировать признаки и обобщать картинку с двух глаз (бинокулярное зрение). V3 распознает цвет и выполняет сегментацию изображения. V4 определяет простые геометрические фигуры. V5 отвечает за распознавание движений. V6 группирует признаки со всего изображения. В V7 происходит распознавание сложных геометрических фигур, а также распознавание человеческих лиц.

Исследования, проведенные Хьюбелом и Визелем, легли в основу моделей распознавания визуальных образов таких, как неокогнитрон и сверточные нейронные сети, которые и будут использованы в дальнейшем.

В мозге присутствует обратная связь от высоких уровней к низких (от V4 к V2 и V1). Подобные обратные связи также находят свое отражение в искусственных нейронных сетях, которые комбинируют сверточные нейронные сети и рекуррентные нейронные сети, например, ResNet [8].

Сигналы с V1 и V2 параллельно обрабатываются несколькими путями (дорсальный путь и вентральный путь). Дорсальный путь начинается с V1, проходит в V2 и заканчивается в V5 и V6. Этот путь связан с локализацией видимых объектов

и движением глаз. Вентральный путь начинается с V1, направляется в V2 и заканчивается в V4. Этот путь связан с распознаванием форм объектов, представлением об объекте и долговременной памятью [60]. Подобная параллельная обработка применяется в сверточных сетях, состоящих из модулей, например, GoogLeNet.

Вторая волна популярности нейронных сетей относится к периоду 1980–1995 годов, которая связана с движением под названием коннекционизм, или параллельная распределенная обработка [58]. Коннекционизм возник как часть научного направления когнитивистики – подхода к изучению процесса познания, который включает в себя разные дисциплины: символьная лингвистика, нейрофизиология, психология, теория познания. Коннекционисты изучать модели познания на основе нейронов, описанные еще Дональдом Хеббом. Идея коннекционизма заключается в описании поведения человека с помощью искусственных нейронных сетей, объединяя простые блоки (нейроны) в различных комбинациях.

Благодаря коннекционизму появилось две важные концепции нейронных сетей: распределенное представление, которое заключается в взаимосвязи одного входа с несколькими признаками, и алгоритм обратного распространения ошибки, который также будет использован ниже [58].

В 1980 году Фукушима предложил архитектору нейронной сети под названием «неокогнитрон», предназначенной для распознавания образов. Эта архитектура основана на работе Хьюбела и Визеля, которые делили клетки первичной зрительной коры на простые и сложные. Простые клетки (нейроны S-типа) реагирует на различные полосы, участки света, границы объекта. Сложные клетки (нейроны C-типа) сигнализируют о наличии признака, полученного из простых клеток, независимо от их местоположения. Тем самым достигается устойчивость к шумам

входного сигнала. Современные сверточные нейронные сети основана на идее неокогнитрона.

В [61,62] рассмотрено применение и проведено исследование неокогнитрона. Неокогнитрон представляет большой интерес для исследования из-за биологического сходства с мозгом, но применить его на практике трудно. Эта разновидность нейронных сетей устойчива к поворотам, наклонам и шумам при применении ее для распознавания символов. Но практическое применение распознавания образов сталкивается с такими проблемами, как разный фон, освещение, разное расстояние от распознаваемого предмета до камеры, разные формы одного и того же класса предметов. И с этими трудностями неокогнитрон не может справиться.

В 1980-х годах Хопфилдом была создана нейронная сеть с обратной связью, которая обладает динамической устойчивостью к искажениям входного сигнала [59].

В 1982 году Кохоненом была опубликована работа по самоорганизующейся карте – разновидность нейронной сети, позволяющая уменьшить размерность пространства входных данных, которая применяется для задачи кластеризации [59].

В 1983 году Барто, Саттон и Андерсон опубликовали работу по обучению с подкреплением [59]. Агент-программа должна путем взаимодействия со средой-учителем научиться взаимодействовать с этой средой и решать определенные задачи. В настоящее время этот вид обучения активно применяется в различных областях, в том числе и в робототехнике.

В 1986 году Румельхартом, Хинтоном и Вильямсом был разработан алгоритм обратного распространения ошибки [59]. Алгоритм заключается в том, что на вход нейронной сети подается изображение, сеть рассчитывает выходной сигнал, и вычисляется ошибка – разность полученного сетью сигнала и эталонного выходного сигнала. Ошибка отправляется в начало сети и коэффициенты нейронов, учитывая эту ошибку, пересчитываются. Тем самым достигается высокая точность классификации

эталонных сигналов. Этот алгоритм является основным для обучения нейронных сетей и по сей день.

В 1989 году Ли Кун предложил архитектуру сверточной нейронной сети (LeNet), предназначенной для распознавания рукописных цифр. Сверточная нейронная сеть – сеть, представляющая собой набор фильтров (сверток), которые путем обучения сети умеют обнаружить от простых линий и закорючек (первые слои сети) до более сложные абстракций, частей изображения (последние слои). LeNet легла в основу всех современных сверточных нейронных сетей, в частности нейронной сети AlexNet для классификации 1000 различных категорий предметов, созданной Хинтоном и Крижевским в 2012 году.

Третья волна популярности нейронных сетей (глубокое обучение) началась примерно в 2006 году [6], когда наступила эра «больших данных». Человечество накопило большое количество информации о мире и появились мощные компьютеры, способные эти данные обработать. Люди производят огромное количество данных каждых данных. Большое количество находятся в открытом доступе в интернете, что позволяет использовать эти данные для своих исследований. Появились различные наборы данных (например, ImageNet, который содержит 14 млн. размеченных изображений, то есть изображений, для которых известно, что на них находится). Третья волна популярности длится до сих пор.

Сверточные сети – это разновидность нейронных сетей, которые состоят из фильтров (сверток). Каждый такой фильтр путем обучения нейронной сети учится отличать от различный линий и закорючек (первые слои) до определенных образов (последние слои) [63].

Первые сверточные нейронные сети были придуманы и применены Ли Кунов в 1989 году [5] для задачи распознавания рукописных символов (LeNet). Но на тот

момент времени сеть LeNet не получила широкой популярности, так как не было достаточных вычислительных мощностей для ее работы.

Вновь к сверточным нейронным сетям вернулись в 2012 году, когда команда Алекса Крижевского заняла первое место в соревновании ILSVRC-2012 [5]. Причем по точности распознавания их решения значительно опередило лучшее решение 2011 года соревнования ILSVRC-2011. Сверточная сеть Алекса Крижевского называется AlexNet [6], и она очень похожа на LeNet.

В дальнейшем на основе сети AlexNet была создана сеть Network-in-Network (NiN) [64], в которой стали применяться маленькие фильтры размером 1x1 и сама сеть состоит из последовательно соединенных нескольких сетей, что увеличило правильность распознавания. Дальнейшее развитие состоит в создании сети VGG [7], которая также структурирована из маленьких фильтров (3x3 и 1x1) и еще большего числа слоев. В дальнейшем идею увеличения количества слоев развила команда Google, которая создала сеть GoogLeNet [65]. В этой сети также применяются маленькие фильтры 3x3 и 1x1 и на основе идеи сети NiN сеть GoogLeNet состоит из последовательно соединенных маленьких сетей (модулей), которые называются Inception.

Такие глубокие сети даже стали отдельным направлением в машинном обучении и стали называться глубоким обучением (Deep Learning) [66]. Эти сети показывают высокий процент правильного распознавания, но скорость их работы достаточно низкая. В связи с развитием мобильных технологий и вычислений на мобильных устройствах (смартфонах) разработчики нейронных сетей захотели перенести свои нейронные сети на смартфоны. Поэтому были созданы сети SqueezeNet [67] и MobileNet [9]. В SqueezeNet содержится большое количество слоев, состоящих их маленьких сверток, но при этом количество параметров в несколько раз меньше (следовательно, количество вычислений меньше), чем у AlexNet, а процент

правильной классификации такой же, как у AlexNet. В MobileNet также, как и в SqueezeNet большое количество слоев с маленькими свертками, но количество параметров еще меньше, чем в SqueezeNet, и эта нейронная сеть работает быстро в смартфонах.

Нами проведены исследования [68] по выбору нейронной сети среди AlexNet, SqueezeNet и MobileNet, так как у этих сетей скорость работы на оборудовании с ограниченным ресурсом (например, RaspberryPi) удовлетворяет поставленной задаче (время обработки меньше 1 секунды).

2.2 Процесс обучения сверточной нейронной сети

Для Индустрии 4.0 важно увеличение количество предприятий по сортировке мусора с последующим увеличением количества данных для обучения модели, что положительно сказывается на качестве модели. Для быстрого роста таких предприятий одним из значимых факторов является низкая стоимость используемого оборудования, а это предполагает ограничении вычислительных возможностей. Также такое оборудование должно быть способно запускать программы с нейронной сетью. По этим обозначенным причинам в качестве оборудования, на котором будет запускаться нейронная сеть для задач СМ, может быть выбран, например, микрокомпьютер RaspberryPi.

2.2.1 Описание экспериментальной установки

Обучение сетей производилось на графической карте Nvidia GeForce GT 740M, состоящей из 384 ядер с тактовой частотой 993 МГц и графической памятью 2048 МБ.

Фреймворк, который был выбран для обучения, – Caffe, так как это популярный фреймворк с большим сообществом разработчиков [69] и модель, которая получается в результате обучения, легко перенести с одного компьютера на другой и которую можно запустить на CPU даже если обучение производилось на GPU. Для запуска моделей на RaspberryPi используется модуль dnn из OpenCV. OpenCV скомпилирован с использованием NEON и VFPV3, что увеличивает скорость работы нейронной сети на RaspberryPi [70]. Предварительная проверка обученной сети производилась на компьютере с центральным процессором Intel Core i7-3610QM с 8 ядрами, базовой тактовой частотой 2,3 ГГц и оперативной памятью 8 ГБ. Проверка обученной сети производилась на RaspberryPi 3 с центральным процессором Broadcom BCM2837 с 4 ядрами, частотой 1,2 ГГц и 1 ГБ оперативной памяти.

Обучение производилась на 500 фотографиях бутылок и 500 фотографиях банок, которые были скачаны из поисковых запросов Google, а также в качестве «прочего мусора» использовались 10000 картинок из базы данных UKBench [71]. Бутылки и банки в обучающей выборке в основном расположены вертикально, горлышком вверх и не мятые. Проверка, которая проводится во время обучения, осуществляется на проверочной выборке, состоящей из 150 фотографий бутылок, столько же банок и 2000 фотографий «прочего» мусора. Эта выборка состоит из фотографий, которых нет в обучающей выборке. Небольшая выборка для тестирования состоит из 10 фотографий бутылок, столько же банок и столько же «прочего» мусора. Эта выборка содержит фотографии бутылок и банок, горлышки которых расположены и вертикально вниз, и вертикально вверх, а также содержит и мятые, и не мятые банки и бутылки.

Для обучения использовалась технология «передачи знания» (Transfer Learning) [72]. Суть этой технологии состоит в том, что обучение нейронных сеть производится не «с нуля», то есть не со случайных значений весов, а с уже обученной

модели. Модели могут быть обучены как на наборе фотографий ImageNet [73], так и на других наборах фотографий. Без использования технологии Transfer Learning обучать нейронные сети пришлось бы на гораздо большем наборе данных и такое обучение длилось бы несколько недель. Обученные модели AlexNet, SqueezeNet получены из Model Zoo [74], MobileNet – из [75].

Transfer Learning подразделяется на два вида [72]:

1. Использование сверточной сети для извлечения признаков. Все веса обученной сети переносятся в новую сеть и остаются в том виде, в котором они есть, но последний слой (классификатор) отбрасывается, и создается новый классификатор, который обучается «с нуля» на новом наборе данных.

2. Точная настройка (fine tuning) сверточной сети. Все веса обученной сети переносятся в новую сеть и корректируются с учетом нового набора данных. То есть обучение производится не со случайного состояния, а с сети, которая уже умеет отличать объекты какого-либо набора данных.

2.2.2 Результаты эксперимента для обученной нейронной сети для распознавания пластиковых и алюминиевых банок

В Caffe процесс обучения определяется количеством итераций. Количество итераций – это количество частей обучающей выборки (batch), на которых проведено обучение [76]. Для обучения разных сетей применялся разный размер этих частей из-за ограничений памяти видеокарты (чем больше этот размер, тем больше требуется памяти). На наш взгляд более показательным является количество эпох, которое показывает сколько полных обучающих выборок прошло через процесс обучения. Размер части выборки влияет на время, которое уходит у нейронной сети на обработку выборки (чем больше этот размер, тем больше время), влияет на разброс процентов

правильного распознавания (чем меньше этот размер, тем больше разброс, чем больше этот размер, тем более плавное происходит увеличение процента правильного распознавания), а также нейронная сеть при слишком маленьком размере части выборки может не обучиться, так как не будет постепенного увеличения процента правильного распознавания, а будет сильный разброс процентов от итерации к итерации [77].

В таблице 2.1 указаны размеры частей, которые использовались для обучения по каждой сети. Столбцом «Один слой» отмечена сеть, у которой обучен только классификатор. Столбцом «Вся сеть» отмечена сеть, у которой обучены все слои.

Таблица 2.1

Размеры частей выборки по каждой сети

	AlexNet		SqueezeNet		MobileNet	
	Один слой	Вся сеть	Один слой	Вся сеть	Один слой	Вся сеть
Размер частей	64	64	32	32	8	8

Для проверки сети брались модели, которые достигали максимума правильности распознавания для проверочной выборки. Этот максимум достигается для разных нейронных сетей по-разному, но судя из опыта после 1000 итерации процент правильного распознавания практически не меняется, поэтому количество итераций обучения для всех моделей определяется 1000. На рисунке 2.1 изображен график изменения правильности распознавания для проверочной выборки нейронных сетей AlexNet, SqueezeNet и MobileNet при точной настройке всей сети.

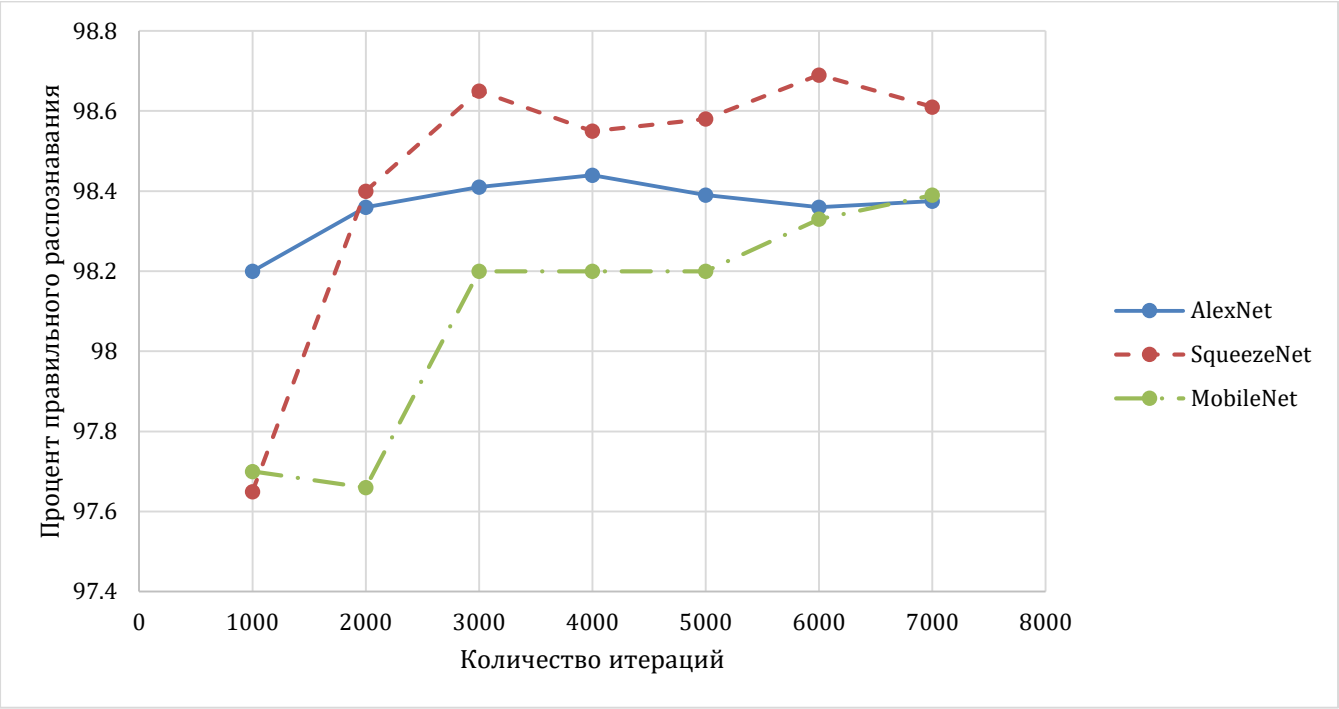

Рисунок 2.1 – Зависимость точности от количества итераций (AlexNet, SqueezeNet и MobileNet)

Как видно из рисунка 2.1, сеть AlexNet достигла процента правильного распознавания примерно в 98,2% за 1000 итераций. Дальнейшие итерации меняют процент правильного распознавания в пределах 0,1%. Сеть SqueezeNet достигла процента правильного распознавания примерно в 97,6% за 1000 итераций, далее 2000 итерация увеличила процент правильного распознавания примерно на 0,8%. 3000 итерация увеличила процент правильного распознавания примерно на 0,2%. Дальнейшие итерации меняют процент правильного распознавания в пределах 0,1%. Сеть MobileNet достигла процента правильного распознавания примерно в 97,7% за 1000 итераций, далее 2000 итерация уменьшила процент правильного распознавания примерно на 0,05%. 3000 итерация увеличила процент правильного распознавания примерно на 0,5%. 4000 и 5000 итерации почти не меняют процент правильного распознавания. 6000 итерация увеличила процент правильного распознавания

примерно на 0,15%. Дальнейшие изменение числа итерации увеличивает процент правильного распознавания в пределах 0,1%.

В таблице 2.2 показано количество итераций, на которой достигается максимум процента правильного распознавания по каждой сети.

Таблица 2.2

Количество итераций, на которых достигается максимум точности

	AlexNet		SqueezeNet		MobileNet	
	Один слой	Вся сеть	Один слой	Вся сеть	Один слой	Вся сеть
Итерация	4000	4000	1000	6000	2000	7000

В таблице 2.3 показано максимальные проценты правильного распознавания по каждой сети для проверочной выборки.

Таблица 2.3

Точности распознавания на проверочной выборке

	AlexNet		SqueezeNet		MobileNet	
	Один слой	Вся сеть	Один слой	Вся сеть	Один слой	Вся сеть
Процент правильного распознавания	97,84	98,44	98,11	98,64	97,85	98,39

Как видно из таблицы 2.3 максимальный процент правильного распознавания для проверочной выборки достигается сетью SqueezeNet со всеми обученными слоями и равен 98,64%.

В таблице 2.4 показаны результаты распознавания по каждой сети для небольшой выборки тестирования. Столбцом «Прав» отмечено, правильно ли

нейронная сеть распознавала образец (если 1, то правильно, если 0, то неправильно). Столбцом «%» показан процент уверенности нейронной сети в том, к какому классу она отнесла образец.

Таблица 2.4

Точности распознавания на небольшой выборке тестирования

		AlexNet				SqueezeNet				MobileNet			
		Один слой		Вся сеть		Один слой		Вся сеть		Один слой		Вся сеть	
		Прав.	%	Прав.	%	Прав.	%	Прав.	%	Прав.	%	Прав.	%
Бутылки	Образец 1	1	100	1	100	1	100	1	100	1	100	1	100
	Образец 2	1	47	1	59	1	77	1	96	0	47	1	49
	Образец 3	1	100	1	100	1	100	1	100	1	98	1	100
	Образец 4	1	99	1	100	1	77	1	90	0	70	1	83
	Образец 5	0	57	0	100	0	49	0	81	0	45	0	88
	Образец 6	1	100	1	100	1	100	1	100	1	96	1	100
	Образец 7	0	100	0	100	0	100	0	99	0	40	0	78
	Образец 8	1	98	1	100	0	74	1	100	1	95	1	100
	Образец 9	1	100	1	100	1	99	1	100	1	100	1	100
	Образец 10	1	63	1	75	1	99	0	93	0	65	1	85
Банки	Образец 11	1	100	1	100	1	96	1	100	1	97	1	100
	Образец 12	1	99	1	100	1	90	1	100	1	100	1	100
	Образец 13	1	97	1	98	0	53	1	69	1	84	1	80
	Образец 14	1	75	1	97	1	98	1	100	1	95	1	100
	Образец 15	1	100	1	100	1	87	1	100	1	98	1	72
	Образец 16	1	98	1	100	1	80	1	100	1	100	1	100
	Образец 17	1	95	1	100	1	99	1	100	1	98	1	100
	Образец 18	1	99	1	100	1	100	1	100	1	100	1	100
	Образец 19	1	97	1	100	1	91	1	100	1	98	1	100
	Образец 20	0	99	0	99	0	100	0	98	1	100	1	87
Мусор	Образец 21	1	99	1	100	1	100	1	100	1	100	1	100
	Образец 22	1	100	1	96	1	99	1	99	1	93	1	98
	Образец 23	1	100	1	100	1	100	1	100	1	99	1	100
	Образец 24	1	100	1	100	1	100	1	100	1	100	1	100
	Образец 25	1	99	1	100	1	100	1	100	1	95	1	98
	Образец 26	1	99	1	100	1	100	1	100	1	99	1	100
	Образец 27	1	100	1	100	1	100	1	100	1	99	1	100
	Образец 28	1	100	1	100	1	100	1	100	1	93	1	100
	Образец 29	1	100	1	100	1	100	1	100	1	100	1	100
	Образец 30	1	100	1	100	1	100	1	100	1	100	1	100
Точность распознавания		90%		90%		83%		87%		83%		93%	

Как видно из таблицы 2.4 самый высокий процент правильного распознавания для небольшой тестовой выборки у сети MobileNet со всеми обученными слоями.

В таблице 2.5 описаны характеристики образцов. «+» отмечено, если образец обладает свойством, указанным в названии столбца, «-» – если не обладает.

Таблица 2.5

Свойства образцов небольшой выборки тестирования

Образцы	Горлышком вверх	Мятый
Образец 1	+	-
Образец 2	-	-
Образец 3	+	+
Образец 4	-	+
Образец 5	-	-
Образец 6	+	+
Образец 7	-	-
Образец 8	+	-
Образец 9	+	-
Образец 10	-	-
Образец 11	+	-
Образец 12	-	-
Образец 13	+	+
Образец 14	+	+
Образец 15	+	+
Образец 16	+	+
Образец 17	+	-
Образец 18	-	-
Образец 19	+	+
Образец 20	+	-

Как видно из таблицы 2.4 и таблицы 2.5 ни одна из рассмотренных сетей не смогла распознавать образец 5 и 7, которая является перевернутой не мятой пластиковой бутылкой.

В таблице 6 показано время распознавания одного изображения по каждой сети на компьютере с CPU, характеристики которого указаны выше, и на RaspberryPi.

Таблица 2.6

Время распознавания одного изображения

	AlexNet		SqueezeNet		MobileNet	
	Один слой	Вся сеть	Один слой	Вся сеть	Один слой	Вся сеть
Время/изображение на CPU (мс)	174	181	373	400	118	117
Время/изображение на RaspberryPi (мс)	887	900	323	317	748	725

Как видно из таблицы 2.6 время распознавания у AlexNet и у MobileNet на RaspberryPi увеличилось в несколько раз, а у SqueezeNet – уменьшилось. Скорее всего, это вызвано хорошей оптимизацией OpenCV под процессоры ARM и небольшим количеством выходов сверток SqueezeNet (максимальное количество выходов у SqueezeNet равно 256, у MobileNet – 1024, у AlexNet – 384, также у AlexNet есть два полносвязных слоя с количеством выходов 4096, которые также могут увеличивать время обработки картинки).

Таким образом, самый высокий процент правильного распознавания среди указанных в таблице 2.4, имеет сеть MobileNet при настройке всех слоев. Минимальное время, необходимое для обработки одного изображения на RaspberryPi среди времен, указанных в таблице 2.6, имеет сеть SqueezeNet.

Максимальная точность распознавания пластиковых бутылок и алюминиевых банок достигается с помощью MobileNet. Поэтому в качестве основной сети в дальнейших исследованиях выбрана MobileNet. В дальнейшем необходимо решить вопросы по увеличению точности выбранной нейронной сети MobileNet с помощью увеличения обучающей выборки методами аугментации данных.

2.3 Увеличение точности распознавания выбранной нейронной сети

2.3.1 Описание метода повышения точности с помощью аугментации

Предлагается метод повышения точности математической модели классификации изображений с помощью аугментации [6]. Этот метод производит поиск оптимальных параметров различных видов преобразования исходных изображений. Параметры и вид преобразования выбираются таким образом, чтобы увеличить точность математической модели классификации изображений на максимальную величину по сравнению с исходной моделью.

Обозначим за $\mathcal{T} = \{T_1, \dots, T_N\}$ множество преобразований исходных изображений. Множество параметров соответствующих преобразований обозначим за $\mathcal{P} = \{P_1, \dots, P_N\}$. Выбор оптимального преобзраования выборки изоражений и соответстующих параметров выполняется с помощью минимизации функционала ошибки классификации изображений:

$$P_{opt}, T_{opt} = argmin\, Q(\mathcal{P}, \mathcal{T}) \tag{3.1}$$

где P_{opt} – оптимальные параметры выбранного оптимального преобразования T_{opt}, Q – функционал ошибки классификации обученной нейронной сети (НС) на общей тестовой выборке.

Для каждого из преобразований получаем несколько преобованных выборок, количество которых равно мощности соответсвующего множества параметров. После получения преобразованных выборок производится обучение нейронной сети, затем выполняется оценка полученных обученных нейронных сетей на общей тестовой выборке. После чего с помощью (3.1) выбирается вид преобзоравания и

соотвествующие параметры. На рисунке 2.2 показана схема выбора оптимальных параметров и преобразованй выборки с помощью предлагаемого метода.

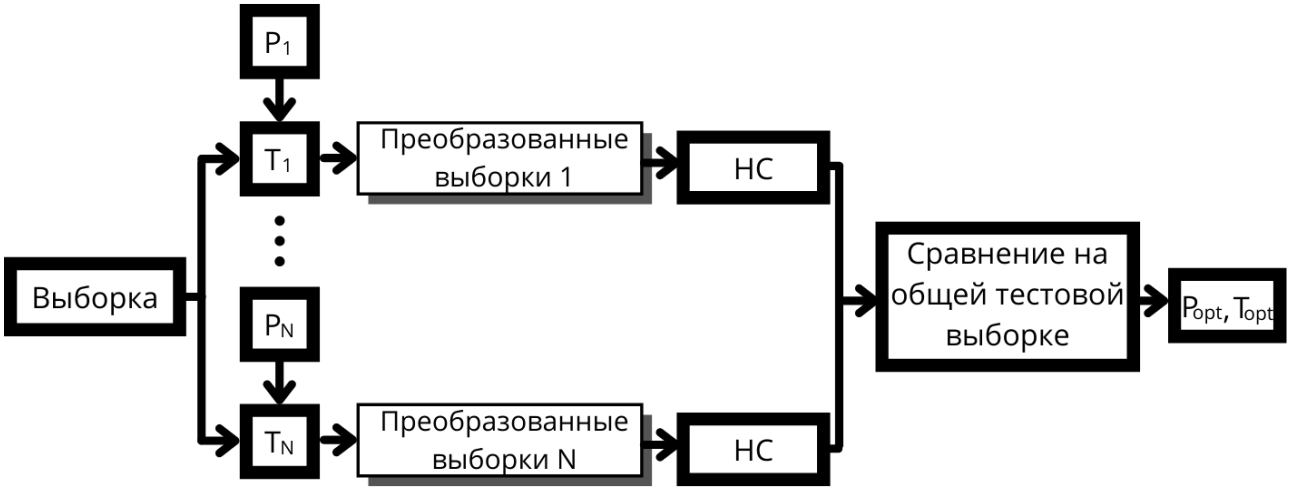

Рисунок 2.2 – Схема предлагаемого метода повышения точности на основе аугментации

Преимущества предлагаемого метода повышения точности математической модели классификации изображений:

1. Параллельное создание обработанных (аугментированных) выборок изображений с различными параметрами аугментации.

2. Параллельное обучение нейронных сетей для различных обработанных выборок изображений.

3. Кэширование (хранение) полученных преобразованных выборок.

4. Децентрализованное хранение обработанных выборок изображений.

Отмеченные преимущества разработанного метода повышения точности позволяют повысить скорость выбора оптимальных параметров аугментации и обучения нейронных для обработанной выборки изображений. Рассмотрим применение предложенного метода для увеличения точности распознавания на основе нейронной сети MobileNet

2.3.2 Проведение экспериментов с предлагаемым методом повышения точности нейронной сети

Обученная нейронная сеть MobileNet достигла 98% правильной классификации на проверочном наборе данных, состоящей из бутылок, банок и прочих предметов. Но после проверки этой сети в режиме онлайн с использованием камеры были обнаружены требования, которые предъявляет нейронная сеть для точной классификации к распознаваемому объекту. Если эти требования не выполняются, то точность распознавания падает. Поэтому необходимо найти способ дообучения нейронной сети с учетом возможного изменения характеристик объекта или его размещения.

2.3.2.1 Обзор методов аугментации

Требования, которые предъявляет нейронная сеть для высокого процента правильной классификации состоит в следующем:

1. Распознаваемый предмет должен располагаться как можно ближе к камере, чтобы он занимал как можно больше пространства на изображении.

2. Предмет должен располагаться вертикально.

3. Предмет должен равномерно освещаться, и лампа, которая освещает предмет, не должна мерцать (то есть недопустимо использование люминесцентной лампы в качестве освещения).

Чтобы обеспечить эти требования, была применена технология увеличения количества данных (data augmentation). Для увеличения количества данных к каждому изображению из обучающей выборки применяется какое-либо преобразование. Data

augmentation успешно была применена в статьях [6,7,78–80], что привело к увеличению процента правильной классификации.

Причина этих требований нейронной сети, обозначенных выше, заключается в переобучении (overfitting) нейронной сети. Проблема переобучения заключается в том, что нейронная сеть слишком «привязывается» к обучающей выборке и не может обобщить знания о тех предметах, которые она должна классифицировать. А причина, почему обученная сеть дает 98% правильной классификации (что является высоким процентом), может заключаться в том, что проверочная выборка не является репрезентативной, то есть не отражает реальных условий проверки.

Известны различные методы "атак" на нейронные сети. В [81] описан способ обмана нейронной сети путем изменения оттенка и насыщенности изображения, что приводит к неправильной классификации распознаваемого изображения и снижению правильности классификации до 90%.

В [82–84] описаны способы «атаки» на нейронные сети путем добавления шума, когда к распознаваемому изображению прибавляют шум, из-за которого изображение меняется так, что человек невооруженным глазом заметить изменение не может. Но нейронная сеть из-за этого шума неправильно классифицирует изображение. И IBM выпустила даже инструмент [85] для предотвращения подобных "атак".

Нейронная сеть, которая была обучена на начальном обучающем наборе фотографий далее в работе будет называться базовой нейронной сетью.

Увеличение обучающей выборке производилось несколькими способами: поворот, отражение по горизонтали (horizontal flipping), гауссовский шум, отступ (padding), удаление части изображения (erasing), вырезание части изображения (cropping), масштабирование (scale). Для получения увеличенной выборки применялись выше обозначенные преобразования с различными вариантами входного параметра (по три для каждого преобразования). После применения каждого

преобразования были получены 7 групп обучающих выборок. В каждой группе находится по три выборки, полученные преобразованиями с разными параметрами. И каждая выборка была в два раза больше относительно начальной.

2.2.2.2 Выбор методов аугментации

Проверка всех обученных сетей в экспериментах на увеличенной выборке производилось на 700 изображениях пластиковых бутылок, алюминиевых банок и прочих предметов, которые были сфотографированы с использованием различных камер, разным фоном, разным освещением (естественное освещение, с помощью светодиодной ламп и с помощью мерцающей люминесцентной лампы), разным расстоянием от камеры до предмета и разным углом наклона между камерой и предметом. Для каждой группы преобразования с разными параметрами были выбраны те преобразования с такими параметрами, которые дают самый высокий процент правильной классификации на проверочной выборке.

У экспериментов по каждому виду преобразования использована следующая структура: описание преобразования, описание входного параметра преобразования, таблица результатов проверки для каждого входного параметра, выбор входного параметра, который дает наибольший процент правильной классификации. Для базовой сети и отражения по горизонтали будет представлен только один результат проверки, так как у них нет параметров.

а) Исходное изображение поворачивается на заданный угол вокруг центра изображения. Входным параметром является угол в градусах, и поворот производится на случайный угол от 0° до угла, заданного во входном параметре. В таблице 2.7 показаны результаты экспериментов при повороте изображения на случайный угол.

Таблица 2.7

Результаты проверки поворота

Угол поворота	15 градусов	30 градусов	45 градусов	60 градусов	75 градусов
% правильной классификации	68	66	75	78	65

Наибольший процент правильной классификации для поворота достигается при 60° и равен 78%.

б) Исходное изображение зеркально отражается по горизонтали, то есть поворачивается вокруг оси, проходящей вертикально через центр изображения.

Таблица 2.8

Результаты проверки отражения по горизонтали

Применение отражения	Отражение по горизонтали
% правильной классификации	65

Процент правильной классификации для отражения по горизонтали равен 65%.

в) К исходному изображению добавляется гауссовский («белый») шум и для каждого канала изображения этот шум разный. Входным параметром является стандартное отклонение. В таблице 2.9 показаны результаты экспериментов при добавлении к изображению гауссовского шума.

Таблица 2.9

Результаты проверки гауссовского шума

Стандартное отклонение	0,05	0,10	0,15	0,20
% правильной классификации	62	64	66	61

Наибольший процент правильной классификации для гауссовского шума достигается при 0,15 и равен 66%.

г) Исходное изображение смещается по горизонтали и вертикали на заданную величину. Входным параметром является доля высоты и ширины изображения, и изображение смещается на случайную величину от 0 пикселей до процента, заданного во входном параметре, умноженного на ширину (при смещении по горизонтали) или на высоту (при смещении по вертикали). В таблице 2.10 показаны результаты экспериментов при отступе изображения.

Таблица 2.10

Результаты проверки отступа

Доля высоты и ширины	0,1	0,2	0,3	0,4
% правильной классификации	72	63	73	66

Наибольший процент правильной классификации для отступа достигается при 0,3 и равен 73%.

д) К исходному изображению добавляется прямоугольник с цветом, равным среднему цвету изображений из ImageNet, с заданным размером и случайным местоположением. Входным параметром является доля площади исходного изображения. Случайным образом выбирается площадь от 0,02, как в [80], до

величины, заданной во входном параметре. Также случайным образом вычисляется соотношение сторон прямоугольника от 0,3 до 3,33 согласно [80]. Ширина прямоугольника рассчитывается как корень квадратный из площади, деленной на соотношение сторон, а высота – как корень квадратный из площади, умноженной на соотношение сторон. В таблице 2.11 показаны результаты экспериментов при удалении части изображения.

Таблица 2.11

Результаты проверки удаления части изображения

Доля площади	0,2	0,3	0,4
% правильной классификации	56	66	65

Наибольший процент правильной классификации для удаления части изображения при 0,3 и равен 66%.

е) Из исходного изображения вырезается часть, ограниченная прямоугольником с заданным размером и случайным местоположением. Входными параметрами являются диапазон доли площади исходного изображения. Случайным образом выбирается площадь от первого входного параметра до второго входного параметра с тем же соотношением сторон, как и у исходного изображения. Так как входным параметром является доля вырезаемой площади, то максимальное значение второго входного параметра равно 1,0. В таблице 2.12 показаны результаты экспериментов при вырезании части изображения.

Таблица 2.12

Результаты проверки вырезания части изображения

Диапазон доли площади	От 0,9 до 1,0	От 0,7 до 0,9	От 0,5 до 0,7	От 0,3 до 0,5
% правильной классификации	72	68	62	60

Наибольший процент правильной классификации для вырезания части изображения при доле площади от 0,9 до 1,0 и равен 72%.

ж) Исходное изображение уменьшается до заданных размерах и располагается на черном фоне в случайном местоположении. Входными параметрами являются диапазон доли площади исходного изображения. Случайным образом выбирается площадь от первого входного параметра до второго входного параметра с тем же соотношением сторон, как и у исходного изображения. В таблице 2.13 показаны результаты экспериментов при масштабировании изображения.

Таблица 2.13

Результаты проверки масштабирования

Диапазон доли площади	От 0,7 до 0,9	От 0,5 до 0,7	От 0,3 до 0,5	От 0,1 до 0,3
% правильной классификации	63	68	71	65

Наибольший процент правильной классификации для масштабирования при доле площади от 0,3 до 0,5 и равен 71%.

Таким образом, применяя аугментации указанным выше образом, проведено увеличение точности работы сверточной нейронной сети для классификации пластиковых бутылок, алюминиевых банок и прочего мусора путем увеличения количества изображений в обучающей выборке применением одинарного

преобразования изображения. Для каждого преобразования изображений был выбран вариант с параметрами, при которых процент правильной классификации наиболее высокий. При проверки базовой нейронной сети в «жестких» условиях на изображениях с тем шумом, который дают камеры, и теми полосами, которые дают люминесцентные лампы, а также на изображениях с разным расстоянием от камеры до предмета и разным углом наклона, процент правильной классификации равен 58%. Самый высокий процент был достигнут при применении поворота на случайный угол от 0° до 60° и равен 78%. То есть точность классификации пластиковых бутылок и алюминиевых банок увеличилась на 20%.

2.4 Выводы

Проведены эксперименты между различными архитектурами сверточных нейронных сетей: AlexNet, SqueezeNet и MobileNet. Наибольшая точность на проверочной выборке (93%), которая состоит из 30 образцов бутылок, банок и прочего мусора, достигается с помощью нейронной сети MobileNet. Поэтому все дальнейшие исследования проводятся для архитектуры MobileNet.

Описан разработанный новый метод повышения точности с помощью применения преобразований исходной выборки изображений (с помощью метода аугментации). Для предложенного метода и выбранной нейронной сети проведены эксперименты с различными видами аугментации. Рассмотренные виды аугментации включают поворот исходного изображения на случайный угол, отражение по горизонтали, добавление гауссовского шума, отступ, удаление части изображения, вырезание части изображения и масштабирование. Наибольшая точность (78%) достигается при повороте на случайный угол от 0° до 60°. Определены граничные величины параметров аугментации (от 0° до 60° - поворот исходного изображения на

случайный угол, применение отражения по горизонтали, 0,15 - добавление гауссовского шума, 0,3 - отступ, 0,3 - удаление части изображения, от 0,9 до 1,0 - вырезание части изображения и от 0,3 до 0,5 - масштабирование) при которых достигаются оптимальные возможности распознавания изображения.

В результате применения разработанного метода повышения точности нейронной точность распознавания обученной базовой сети MobileNet была увеличена на 20%.

Глава 3. Увеличение точности и скорости распознавания выбранной нейронной сети

Помимо метода преобразования исходной выборки существуют и другие методы повышения точности нейронной сети. Одним из таких методов является выбор оптимальных гиперпараметров, которые помимо увеличения точности позволяют также увеличить скорость обучения нейронной сети. Важным является и метод оптимизации нейронной сети - оптимизация вычислений нейронной сети, которая не меняет ее точность, но увеличивает скорость вычислений внутри нейронной сети. Рассмотрим применение указанных методов к решаемой задаче.

3.1 Оптимизация гиперпараметров

Гиперпараметры – это параметры обучения и архитектуры нейронной сети. В диссертации для оптимизации гиперпараметров будут использоваться только параметры обучения нейронной сети, а сама архитектура остается без изменений. Настройка гиперпараметров позволяет добиться такого обучения нейронной сети, при которой эта сеть не переобучена и не недообучена, а достигает оптимального состояния обучения. Благодаря такому состоянию (с оптимальными гиперпараметрами) получаем наибольшую точность нейронной сети, которая может быть достигнута при меньшем количестве итераций обучения, чем при неоптимальных гиперпараметрах.

3.1.1 Постановка задачи выбора оптимальных гиперпараметров

Оптимизация гиперпараметров применяется для решения различных задач, таких как компьютерное зрение [86,87], робототехника [88,89], обработка естественного языка [90,91] и синтез речи [92].

Проблема выбора оптимальных гиперпараметров известна давно [93-95]. Существующие решения можно разделить по следующим признакам:

1. Количество оптимальных решений.

2. Количество решаемых задач.

3. Количество критериев выбора оптимального решения.

Задача — это набор изображений с количеством классов $N_{classes}$ и количеством изображений N_{images}. Между задачами есть примеры одних и тех же классов, отличие заключается в том, как создаются изображения (разное освещение, фон и используемые камеры). Критерий — это количественная характеристика обучения / оценки нейронной сети по задаче (например, точность, время обработки одного изображения или эпоха, на которой процесс обучения нейронной сети достигает сходимости).

В [96] предлагается метод оптимизации Парето, в котором оптимальное решение дается для нескольких задач одновременно. Этот метод заключается в минимизации взвешенной суммы функций потерь для каждой задачи. В [97] описывается метод оптимизации Парето, который дает оптимальное решение в соответствии с несколькими критериями, основанными на градиентном спуске, и эта оптимизация выполняется в процессе обучения. В [98] поиск оптимального по Парето решения осуществляется по нескольким критериям. В [99] описывает несколько методов многокритериальной оптимизации по Парето. Метод из [93] дает оптимальные гиперпараметры, используя обратное распространение ошибки через

разложение Холецкого. В [100] оптимизация выполняется с использованием случайного выбора гиперпараметров на основе критерия ожидаемого улучшения. [101] предлагает метод гиперпараметрической оптимизации, основанный на случайном поиске в пространстве гиперпараметров. В [102] поиск оптимальных гиперпараметров осуществляется с помощью байесовской оптимизации. В [103] предлагает метод поиска оптимальных гиперпараметров с помощью многозадачной байесовской оптимизации. [104,105] описывают байесовские методы многокритериальной оптимизации.

Рассмотренные существующие методы оптимизации гиперпараметров не позволяют производить оптимизации **одновременно** по разным критериям и по различным задачам. Поэтому необходимо разработать такой метод оптимизации гиперпараметров, которой бы устранял указанный недостаток.

3.1.2 Разработка метода выбора оптимальных гиперпараметров

Предлагаемый оригинальный многозадачный многокритериальный (MTMC) метод основан на оптимизации по Парето. Оптимальность по Парето означает, что невозможно улучшить оптимальное решение по Парето по каким-либо критериям, не ухудшив его хотя бы по одному другому критерию. Таким образом, для определенного набора значений критериев выбранные по Парето решения являются оптимальными. Критерии, наиболее близкие к данным критериям, определяются путем нахождения минимальной взвешенной суммы решений Парето (где значение веса является обратным критерию). Отметим отличительные особенности предлагаемого метода:

1. Оптимизация проводится одновременно по нескольким критериям и нескольким задачам с установкой значимости критериев.

2. Обеспечивается выбор оптимальных гиперпараметров *после* обучения и оценки, что избавляет от необходимости повторно обучать модель.

3. Предлагаемый метод не требует обучения.

В предлагаемом методе модель оценивается на нескольких тестовых наборах (задачах) T. Задача нахождения минимума для задач T известна как минимизация математического ожидания эмпирического риска [106]. Отсюда выбор оптимальных гиперпараметров формализуется следующим образом:

$$\theta = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathbb{E}_{\tau}[L(\theta, \phi)] \tag{3.1}$$

где Θ – множество всех гиперпараметров, θ – выбранные оптимальные гиперпараметры, ϕ – вектор коэффициентов значимости критериев, $L(\cdot)$ – функция оценки модели с заданными гиперпараметрами θ и коэффициентами ϕ, τ – задача, для которой выполняется оптимизация. Предлагаемый метод дает решение задачи (3.1).

Построение критериев производится на основе оптимизации по Парето (3.2):

$$\begin{cases} \forall i \in \{1, \dots, N\}, f_i(\overrightarrow{x_1}) \leq f_i(\overrightarrow{x_2}) \\ \exists j \epsilon \{1, \dots, N\}, f_j(\overrightarrow{x_1}) < f_j(\overrightarrow{x_2}) \end{cases}, \tag{3.2}$$

Согласно (3.1), метод МТМС должен удовлетворять следующим требованиям:

1) метод должен решать задачу минимизации;

2) значимость каждого критерия определяется вектором коэффициентов ϕ (чем выше коэффициент, тем важнее соответствующий критерий).

Введем обозначение тестовой выборки задачи τ:

$$x^i \sim D, i = 1..N_{task} \tag{3.3}$$

где x^i – это i-я тестовая выборка, имеющая распределение D, N_{task} – количество задач.

Перед выбором гиперпараметров для модели M получаем оценочную матрицу для тестовой выборки x^i и заданных критериев:

$$V = M(x^i; \Theta) \tag{3.4}$$

$$M(x^i; \Theta)::(\mathbb{R}^{x_{size}}, \mathbb{R}^{N_{parameter} \times N_{combination}}) \rightarrow \mathbb{R}^{N_{combination} \times N_{criteria}} \tag{3.5}$$

где $M(\cdot)$ – функция модели, которая преобразует заданную выборку x^i и с заданными гиперпараметрами Θ в матрицу оценки V, $N_{criteria}$ – количество критериев, x_{size} – размер тестовой выборки, $N_{parameter}$ – количество гиперпараметров, $N_{combination}$ – количество комбинаций гиперпараметров.

Затем функция L вычисляется для каждой выборки x^i, которая формально описывается следующим образом:

$$L(\cdot; \Theta, \phi) = E(V; \phi) \tag{3.6}$$

$$E(V; \phi)::(\mathbb{R}^{N_{criteria}}, \mathbb{R}^{N_{criteria}}) \rightarrow \mathbb{R}^1 \tag{3.7}$$

Метод МТМС дает оптимальные по Парето решения, в которых выполняются следующие шаги:

Шаг 1. Векторы из оценки V (количество таких векторов $N_{criteria}$) находятся в пространстве заданных критериев.

Шаг 2. Парето дает решения, которые ближе к оптимальным по одному критерию и дальше от оптимальных по другим критериям. Получаем оптимальные по Парето решения $\hat{V} \subset V$ – ближайший Парето-фронт к началу координат пространства критериев:

$$\hat{V} = ParetoFront(V), \hat{V} \in \mathbb{R}^{N_{opt} \times N_{criteria}} \tag{3.8}$$

где N_{opt}– количество оптимальных по Парето решений, $ParetoFront(\cdot)$ – выбор Парето-оптимальных решений с помощью (3.2).

Шаг 3. Оптимальные решения $\hat{v} \in \hat{V}$ масштабируются по каждому критерию до интервала $[0;\ 1]$:

$$\hat{V}_{scaled} = \frac{\hat{V}_i - \hat{v}_{min}}{\hat{v}_{max} - \hat{v}_{min}}, \hat{v}_{max} \in \mathbb{R}^{N_{criteria}}, \hat{v}_{min} \in \mathbb{R}^{N_{criteria}}, i = 1..N_{opt} \quad (3.9)$$

где $\hat{v}_{max}$ – вектор максимальных значений $\hat{V}$для каждого критерия, $\hat{v}_{min}$ – вектор минимальных значений $\hat{V}$для каждого критерия.

Таким образом, оптимальное решение - это решение, ближайшее к началу координат, и если какое-либо решение $\hat{v}$ является началом координат, то оно оптимально для любого ϕ.

Шаг 4. Определить вектор ϕ в пространстве критериев.

Введем вектор оптимального решения, который является серединой отрезка $[0; 1]$ по осям пространства критериев:

$$\phi_{opt} = \left(\forall i: \phi_0 = \ldots = \phi_i = \ldots = \phi_{N_{criteria}} = \frac{1}{2}\right). \quad (3.10)$$

$$\phi = \phi_{opt}, \text{ если } \forall i: \phi_i = 0, \text{ иначе} \phi \in [0; 1]. \quad (3.11)$$

Шаг 5. Необходимо определить, какой гиперпараметр в пространстве критериев ближе к данному критерию. Спроецируем векторы из матрицы $\hat{V}_{scaled}$ на вектор ϕ:

$$\hat{V}_{proj} = \frac{\hat{V}_{scaled}^T \cdot \phi}{\|\phi\|}, \hat{V}_{scaled} \in \mathbb{R}^{N_{opt}}. \quad (3.12)$$

Из (3.10) и (3.12) следует, что если векторы ϕ и ϕ_{opt} коллинеарны, то:

$$\exists \lambda: \phi = \lambda \cdot \phi_{opt} \Rightarrow \sum_i \left[\frac{1}{\phi_i} \cdot \frac{\hat{V}_{scaled_i}}{\|\phi\|}\right] \propto \sum_i \hat{V}_{scaled\,i} = \left\|\hat{V}_{scaled}\right\|_1. \quad (3.13)$$

То есть, в случае равенства всех компонентов ϕ задача минимизации сводится к нахождению минимальной L_1-нормы $\hat{V}_{scaled}$.

Из (3.12) также следует, что если какой-либо компонент вектора ϕ равен нулю, то соответствующий критерий не повлияет на выбор оптимального гиперпараметра. Если все критерии равны нулю, кроме одного, то на выбор оптимальных гиперпараметров будет влиять только критерий с ненулевым компонентом вектора ϕ.

Шаг 6. Находим гиперпараметры θ, при которых достигается минимум вектора $\hat{V}_{proj}$, что эквивалентно поиску минимальной взвешенной суммы значений вектора:

$$\theta = argmin_{\theta}\hat{V}_{proj}. \quad (3.14)$$

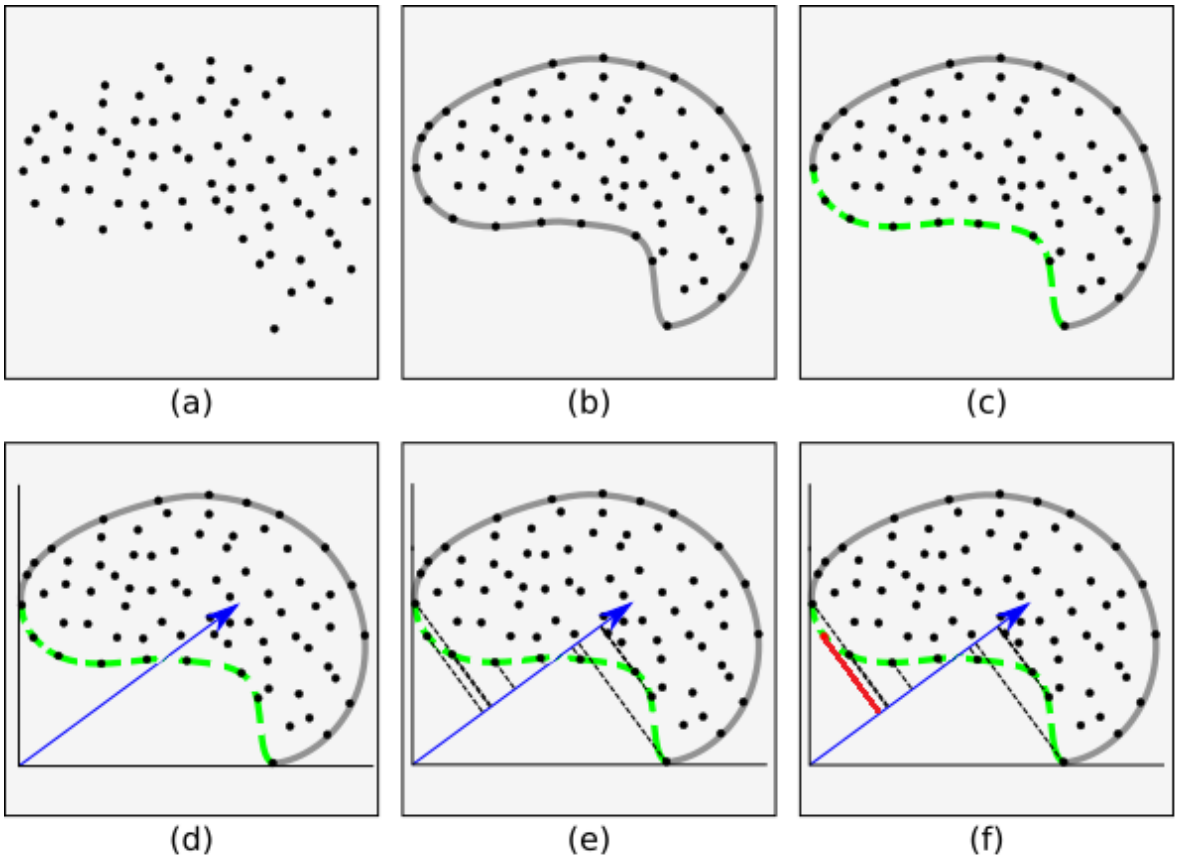

Рисунок 3.1 – Иллюстрация этапов предлагаемого метода MTMC. (a) Оценка всех комбинаций гиперпараметров в пространстве критериев. (b) Получение границы оценки. (c) Получение оптимальных по Парето решений. (d) Определение вектора критериев. (e) Проецирование оптимальных по Парето решений на вектор критериев. (f) Поиск ближайшей проекции на начало координат.

На рисунке 3.1 показаны шаги MTMC для получения оптимального решения для заданных критериев.

На рисунке 3.2 показан пример решения с использованием МТМС для случайных чисел в трехмерном пространстве.

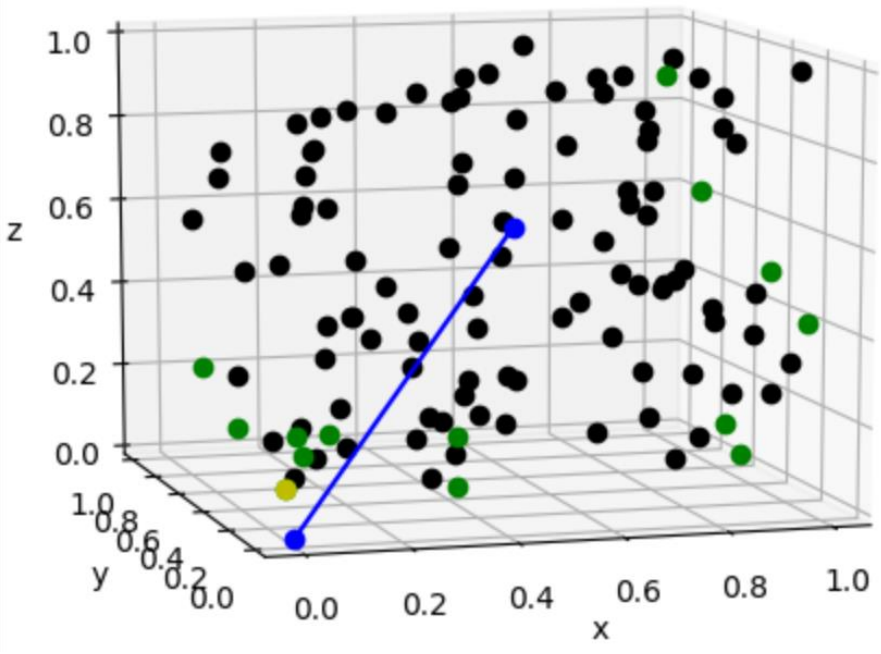

Рисунок 3.2 – Пример решения, полученное с помощью МТМС, зеленые точки обозначают оптимальные решения по Парето, синий вектор – вектор ϕ, желтая точка обозначает оптимальное решение для заданных критериев

Применим разработанный метод МТМС для выбора оптимальных гиперпараметров к задаче классификации бытовых отходов. С этой целью проведем ряд экспериментов.

3.1.3 Проведение экспериментов для разработанного метода МТМС

Получим оценочную матрицу Vдля модели M. Затем для различных комбинаций компонентов ϕ выбираются оптимальные гиперпараметры с помощью МТМС.

Разработанный метод MTMC применяется для решения задачи классификации изображений. Решаемая проблема, описанная в [107], заключается в выборе таких гиперпараметров, которые позволяют достичь максимальной точности классификации среди нескольких задач. Каждая задача состоит из нескольких изображений одного из двух объектов: пластиковой бутылки и другого объекта. Задачи отличаются тем, при каких условиях были созданы изображения, а именно - освещением, фоном и используемыми камерами.

Поиск оптимальных гиперпараметров проводился среди двух популярных методов обучения: изменение скорости обучения на основе эпохи [108] $lr = base\_lr \cdot lr\_decay^{epoch}$ (где $base\_lr$- начальная скорость обучения, lr_decay- коэффициент уменьшения скорости обучения, $epoch$- количество эпох) и циклического обучения [109]. В циклическом обучении есть три способа изменить скорость обучения:

1. *triangular* - это фиксированная начальная скорость обучения ($base\_lr$), максимальная фиксированная скорость обучения (max_lr), скорость обучения увеличивается с $base\_lr$ до max_lr и линейно уменьшается от max_lr до $base\_lr$.

2. *triangular2* - это фиксированная начальная скорость обучения ($base\_lr$), максимальная скорость обучения (max_lr), скорость обучения, как в треугольнике, изменяется линейно, но max_lr в процессе обучения уменьшается вдвое.

3. *exp_range* - это фиксированная начальная скорость обучения ($base\_lr$), максимальная скорость обучения (max_lr), скорость обучения также изменяется линейно, но max $\_lr$ в процессе обучения экспоненциально уменьшается.

В обучении на основе эпохи гиперпараметры представляют собой значение начальной скорости обучения ($base\_lr$) и коэффициент уменьшения скорости обучения (lr_decay). В циклическом обучении гиперпараметры - способ изменить скорость обучения ($cyclic\_mode$), значение начальной скорости обучения ($base\_lr$) и максимальной скорости обучения (max_lr).

Для каждой комбинации гиперпараметров обучение проводилось с использованием k-кратной перекрестной проверки (k-fold cross validation) [110] с 10 кратностями (фолдами). Для обучения использовались Keras [111] и TensorFlow [112]. Обучение длилось 15 эпох. Проверка обученной нейронной сети проводилась на $N_{task} = 5$ различных наборах тестовых выборок. То есть $100 \cdot 10 = 1000$ - это количество разных нейронных сетей, проведено проверок нейронных сетей $15 \cdot 1000 = 15000$, получено результатов проверок $15000 \cdot 5 = 75000$.

Среди всех эпох, для каждого фолда и для каждой тестовой выборки выбирается максимальная точность, а также номер эпохи, на которой достигается максимальная точность. Следующие значения вычисляются для каждой тестовой выборки среди фолдов: математическое ожидание и дисперсия ошибки классификации, математическое ожидание и дисперсия номера эпохи, при которой достигается сходимость на тестовой выборке. Получена оценочная матрица V среди всех нейронных сетей с их гиперпараметрами и среди всех тестовых выборок. В Приложении А показана полученная матрица V.

На основе (3.1) для каждого критерия среди всех выборок рассматривается математическое ожидание. То есть для всех тестовых выборок критериями являются: выборочное среднее ошибки классификации, выборочная дисперсия ошибки классификации, выборочное среднее и выборочная дисперсия номер эпохи, при которой достигается сходимость на тестовой выборке. Эти значения являются критериями для оценки гиперпараметров для определенного набора тестов (матрица V из (3.4)) с количеством критериев $N_{criteria} = 4$. V вычисляется из (3.8), после построения Парето-фронта и выполнения оптимизации по заданным критериям получим $N_{opt} = 25$ оптимальных по Парето решения. Оптимальные гиперпараметры, т.е. V, для первого способа обучения представлены в Приложении Б, для второго - в Приложении В.

Вектор оптимального решения согласно (3.10) для $N_{criteria} = 4$ равен $\phi_{opt} = \{0.5;\ 0.5;\ 0.5;\ 0.5\}$. Далее расчеты проводятся согласно (3.9) и (3.12), и для различных ϕ оптимальные решения выбираются согласно (3.6). Эти оптимальные решения представлены в Приложении Г.

Таким образом, получены гиперпараметры, являющиеся оптимальными по всем критериям (при всех равных значениях критериев), а также при задании одних критериев более значимыми, чем другие. Полученные гиперпараметры будут использоваться при обучении для задач по сортировке бытовых отходов.

3.1.4 Сравнение МТМС с существующими методами гиперпараметрической оптимизации

Существующие методы гиперпараметрической оптимизации можно разделить на три категории: поиск по сетке параметров [113], случайный поиск [113] и байесовская оптимизация [114]. Поведем сравнение МТМС с методом случайного поиска и байесовской оптимизация. Для сравнения выбраны именно эти методы из-за их популярности среди методов гиперпараметрической оптимизации.

МТМС основан на переборе по сетке параметров, и этот перебор для каждого набора гиперпараметров выполняется независимо. То есть мы можем запустить поиск для каждого набора гиперпараметров независимо друг от друга на нескольких устройствах параллельно, поэтому при увеличении количества устройств увеличение скорости получения оценок близко к линейному.

Все полученные оценки поступают на центральный сервер, который выполняет сравнение полученных оценок и выбирает наилучший по заданным критериям. Время на отправку полученных оценок и на их сравнение по сравнению с получением оценок пренебрежимо мало.

В случае байесовской оптимизации поиск выполняется последовательно, а также для небольшого пространства поиска гиперпараметров требуется больше итераций для получения оптимальных гиперпараметров. При росте пространства поиска байесовская оптимизация дает оптимум быстрее, чем поиск по сетке параметров. Также для байесовской оптимизации необходимо перед поиском задать критерии поиска.

В МТМС получаем оценки для различных критериев. При многократном обучении нейронной сети процедура обучения следующая: обучаем нейронную сеть с гиперпараметрами по критерию наибольшей точности, затем при дополнении обучающей выборки данными из того же распределения выполняем дообучение нейронной сети с критериям большей скорости. Для МТМС не нужно выполнять поиск гиперпараметров при других критериев. Для байесовской оптимизации требуется заново провести поиск гиперпараметров для критериев большей скорости.

Определим функцию зависимости времени получения оптимальных гиперпараметров для поиска по сетке параметров (МТМС), случайного поиска и байесовской оптимизации. Сделаем допущение, что эта функция линейная и зависит от количества комбинаций гиперпараметров [115], количества устройств, на которых выполняется параллельное обучение нейронных сетей с получением оценок, и количества задач. тогда, функция времени получения оценок для поиска по сетке параметров (МТМС) имеет следующий вид:

$$T_G = \frac{K_G \cdot N_{params} + B_G}{N_{devices}} \cdot N_{tasks} \tag{3.15}$$

где N_{params} – количество комбинаций гиперпараметров, K_G – коэффициент для поиска по сетке параметров, B_G – смещение для поиска по сетке параметров, $N_{devices}$ – количество устройств, на которых независимо друг от друга выполняется обучение нейронных сетей, N_{tasks} – количество задач.

Так как случайный поиск уменьшает количество комбинаций гиперпараметров за счет случайного выбора гиперпараметров [113], то по аналогии с (3.15) выведем функцию времени получения оценок для случайного поиска:

$$T_R = \frac{K_R \cdot N_{iters} + B_R}{N_{devices}} \cdot N_{tasks} \tag{3.16}$$

где N_{iters} – количество итераций, которым ограничивается поиск гиперпараметров ($N_{iters} \leq N_{params}$), K_R – коэффициент для случайного поиска, B_R – смещение для случайного поиска.

Так как байесовская оптимизация выполняется последовательно [116], то на каждом устройстве оптимизация выполняется отдельно и независимо, этот вид оптимизации дает столько наборов оптимальных гиперпараметров, на скольких устройствах оптимизация была запущено, поэтому время работы байесовской оптимизации не зависит от количества устройств. Также для этого вида оптимизации критерии необходимо добавлять перед выполнением оптимизации и количество запусков возрастает во столько раз, для скольких критериев необходимо получить оптимальные гиперпараметры. По аналогии с (3.15) получим время работы байесовской оптимизации:

$$T_B = (K_B \cdot N_{params} + B_B) \cdot N_{criteria} \cdot N_{tasks} \tag{3.17}$$

где K_B – коэффициент для байесовской оптимизации, B_B – смещение для байесовской оптимизации, $N_{criteria}$ – количество различных критериев поиска оптимальных гиперпараметров.

Так как перебор по сетке параметров (МТМС) избыточен по сравнению с байесовской оптимизацией и случайным поиском, то его коэффициент имеет большее значение. Также для небольшого числа гиперпараметров для байесовской оптимизации требуется большее время для получения оценок, чем для перебора по сетке параметров, а для случайного поиска – меньшее количество переборов за счет

задания количества итераций переборов. Отсюда получаем следующую систему уравнений:

$$\begin{cases} K_G \approx K_R > K_B, \\ B_B > B_G \approx B_{R.} \end{cases} \quad (3.18)$$

Можно заметить из (3.15), (3.16) и (3.18), что случайный поиск менее затратный по времени за счет меньшего количества итераций поиска, чем поиск по сетке параметров, но недостатком случайного поиска является наличие дополнительного параметра – количество итераций, а также необходимость задания априорного распределения для гиперпараметров.

На основе (3.15)-(3.18) промоделируем время получения оценок для перебора по сетке параметров, случайного поиска и байесовской оптимизации, задав для примера $K_G = 25$, $B_G = 20$, $K_R = 25$, $B_R = 20$, $K_B = 2$, $B_B = 30$, $N_{criteria} = 4$, $N_{tasks} = 3$. Количество гиперпараметров N_{params} зададим в отрезке от 10 до 100 с шагом 10, количество итераций для случайного поиска N_{iters} зададим в два раза меньше, чем количество гиперпараметров. Для указанных метод оптимизации гиперпараметров получим время как для одного устройства $N_{devices} = 1$, так и для восьми $N_{devices} = 8$. На рисунке 3.3 представлены графики зависимостей времени получения оценок от количества гиперпараметров.

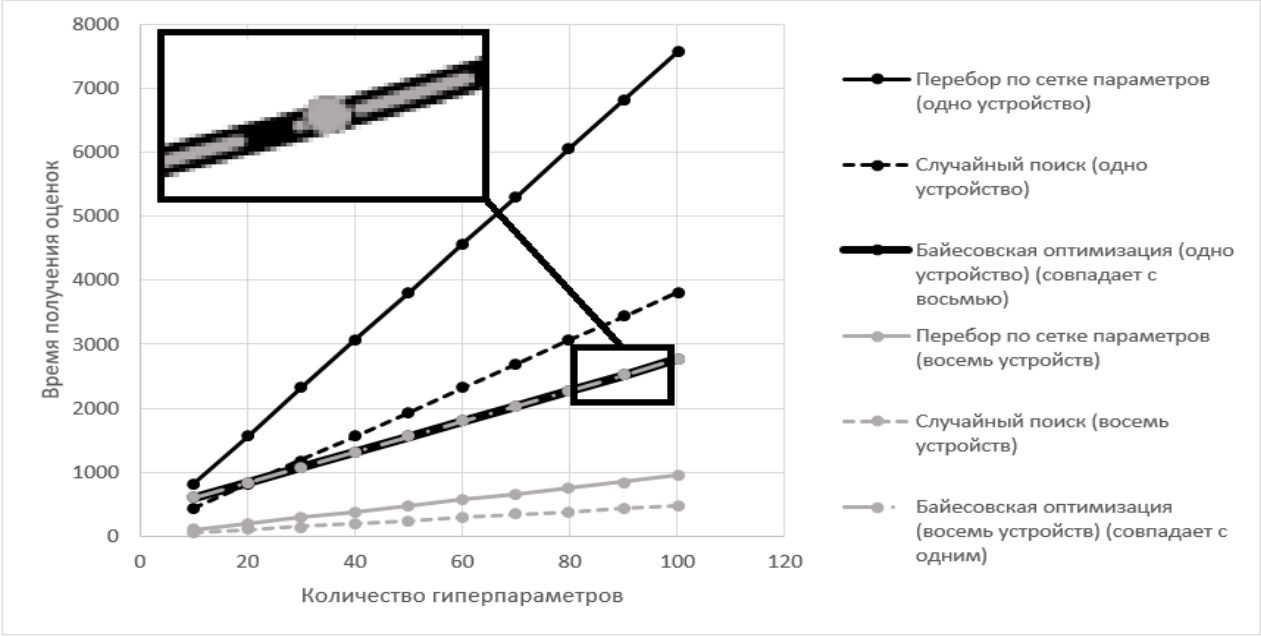

Рисунок 3.3 – График зависимостей времени получения оценок от количества гиперпараметров для перебора по сетке параметров, случайный поиск и байесовская оптимизация для одного и восьми устройств

Как видно из рисунка 3.3, время получения оценок для перебора по сетке параметров (МТМС) больше, чем по остальным рассмотренным методам, но за счет распараллеливания обучения нейронных сетей на нескольких устройствах время получения оценок становится наименьшим среди рассмотренных методов гиперпараметрической оптимизации. Для восьми устройств при заданных коэффициентах время получения оценок МТМС по сравнению со случайным поиском меньше примерно на 300%, с байесовской оптимизацией – на 200%.

Оптимизация гиперпараметров является одним из видов оптимизации. Другим видом оптимизации является изменение вычислений нейронной сети для увеличения ее скорости работы без потери точности. Рассмотрим решение этой задачи по отношению к сортировке отходов на вычислителях с ограниченными возможностями.

3.2 Оптимизация вычислений выбранной нейронной сети

Микрокомпьютер обладает ограниченной вычислительной мощностью. По этой причине к нейронной сети помимо высокой точности предъявляется дополнительное требование - высокое быстродействие и малый объем занимаемой памяти. Проблема заключается в преобразовании обученной нейронной сети таким образом, чтобы ее быстродействие увеличилось, размер модели нейронной сети уменьшился, а точность при этом осталась прежней или незначительно уменьшилась.

3.2.1 Постановка задачи оптимизации вычислений нейронной сети

Необходимо найти такое преобразования весовых коэффициентов нейронной сети, при котором достигается наименьший размер файла модели нейронной сети, время обработки изображения, а точность на тестовой выборке оставалась как можно более близкой к точности до применения преобразования. Для выбора такого преобразования выполняется многокритериальная оптимизация поиска функции преобразования весовых коэффициентов нейронной сети методом квантования. Метод квантования [117] используется из-за возможности его применения для обученной нейронной сети без необходимости заново обучать нейронную сеть.

Для поиска преобразования модели нейронной сети выполняется минимизация следующей оптимизационной функции:

$$L(U,F) = \frac{S(U,F)-S(U_0,F)}{S(U_0,F)} + \frac{T(U,F)-T(U_0,F)}{T(U_0,F)} + \frac{A(U,F)-A(U_0,F)}{A(U_0,F)} = \Delta_r S(U,F) + \Delta_r T(U,F) + \Delta_r A(U,F) \quad (3.19)$$

где F – функция преобразования входных данных в распределение вероятностей принадлежности к определенному классу объекту (такой функцией является

нейронная сеть), U – функция изменения параметров F(такими параметрами являются весовые коэффициенты нейронной сети), U_0 – отсутствие изменения весовых коэффициентов, S – размер файла весовых коэффициентов нейронной сети, T – время обработки одного изображения нейронной сетью, A – точность нейронной сети на тестовой выборке.

Необходимо найти функция изменения весовых коэффициентов U:

$$U = argmin_{U \in \mathcal{U}} L(U, F) \tag{3.20}$$

где $\mathcal{U}$ – множество функций изменения весовых коэффициентов нейронной сети.

Оптимизации функции (3.19) относится к безусловной. Как следует из формулы (3.20), метод решения оптимизационной задачи заключается в поиске функции изменения весовых коэффициентов, при которой значение оптимизационной функции L достигает минимума.

Шаги предлагаемого метода оптимизации вычислений нейронной сети:

Шаг 1. Обучить нейронную сеть для заданной задачи.

Шаг 2. Выбрать множество функций преобразования весовых коэффициентов нейронной сети.

Шаг 3. Для каждой из этих функций произвести преобразование весовых коэффициентов нейронной сети.

Шаг 4. Для преобразованных нейронных сетей вычислить значений функции L по формуле (3.18).

Шаг 5. Выбрать функцию преобразований весовых коэффициентов, которое соответствует наименьшему значению L, то есть решить задачу (3.19).

Для выполнения преобразования весовых коэффициентов предлагается использовать квантование [117], которое применяется в работах [118–122] для сжатия нейронных сетей.

3.2.2 Квантование нейронной сети

Квантование — это уменьшение точности чисел весовых коэффициентов моделей, при котором достигается наименьшая потеря в точности модели. Из [117] квантование вещественного числа в целочисленное производится по следующей формуле:

$$q_3^{(i,k)} = Z_3 + \sum_{i=1}^{N} a^{(i,j)} b^{(j,k)} = Z_3 + \sum_{i=1}^{N} \left(q_1^{(i,j)} - Z_1\right)\left(q_2^{(j,k)} - Z_2\right) \quad (3.21)$$

где q_3 – исходное вещественное число, a – квантованные целочисленные значения, b – весовые коэффициенты квантования, q_1 и q_2 – квантованные целочисленные значения, Z_1, Z_2 и Z_3 – смещение в нулевую точку для q_1, q_2 и q_3 соответственно.

Для квантования использовалась программа TensorFlow Lite [123]. Выборка изображений содержит 1700 изображений бутылок и банок. Все эти изображения разделены по материалу предмета и категории, что в общей сложности составляет пять классов. Вся выборка поделена на обучающую выборку и тестовую в соотношении 80/20. Проверка нейронных сетей производилась на CPU и в качестве входных данных использовалась тестовая выборка. Для визуализации архитектуры нейронной сети использовалась программа Netron [124].

Выполнена оптимизация сети по нескольким видам квантования:

1) квантование динамического диапазона (U_1);

2) квантование с использование репрезентативного набора данных (U_2);

3) целочисленное квантование с использованием репрезентативного набора данных (U_3);

4) квантование во float16 (U_4).

То есть оптимизация целевой функции L производится среди множества $\mathcal{U} = \{U_0, U_1, U_2, U_3, U_4\}$. U_0 соответствует отсутствию квантованию (исходная модель без

преобразований). Далее будут показаны те изменения, которые вносит определенное квантование.

На рисунке 3.4 показаны типы данных для входа исходной нейронной сети до квантования.

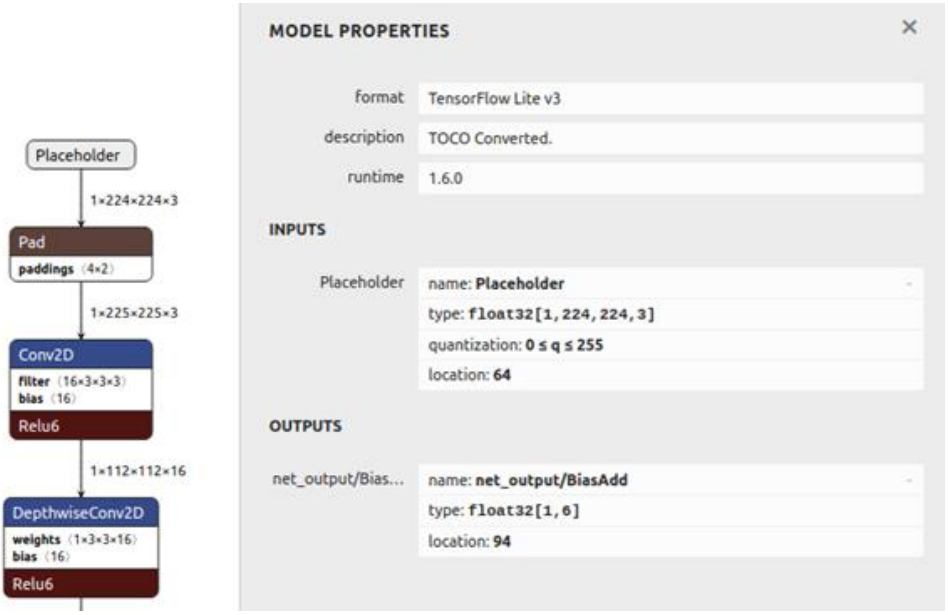

Рисунок 3.4 – Вход исходной нейронной сети

Как видно из рисунка 3.4, на вход полученная нейронная сеть принимает тип данных float32, а также все скрытые слои этой сети тоже используют тип данных float32.

Проведено квантование динамического диапазона. После этого тип данных, в котором хранятся свертки нейронной сети, поменялся с float32 на int8. Также в архитектуру добавляются блоки деквантования, которые меняют тип данных int8 на float32, который используется для вычислений внутри нейронной сети. На рисунке 3.5 показана архитектура квантованной сети с динамическим диапазоном.

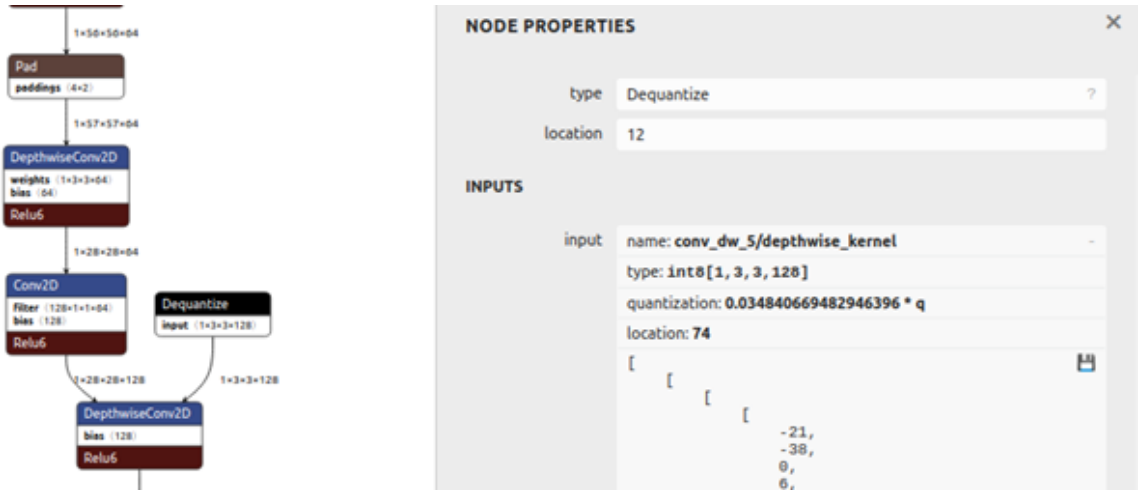

Рисунок 3.5 – Добавление блока деквантования для сверток после квантования динамического диапазона

После применения квантования с использованием репрезентативного набора данных внутри нейронной сети тип данных int8 и в начале архитектуры добавляется блок квантования. На рисунке 3.6 показаны свойства свертки после применения этого типа квантования.

Рисунок 3.6 – Изменение точности чисел сверток после квантования с репрезентативным набором данных

После применения целочисленного квантования с использованием репрезентативного набора данных внутри нейронной сети тот же тип данных int8, что и у квантования, описанного ранее, но отличие заключается в используемом входном типе данных uint8 (только этот тип данных допустим в качестве входа в TPU и микроконтроллеров [125]). На рисунке 3.7 показана характеристика для входного узла нейронной сети.

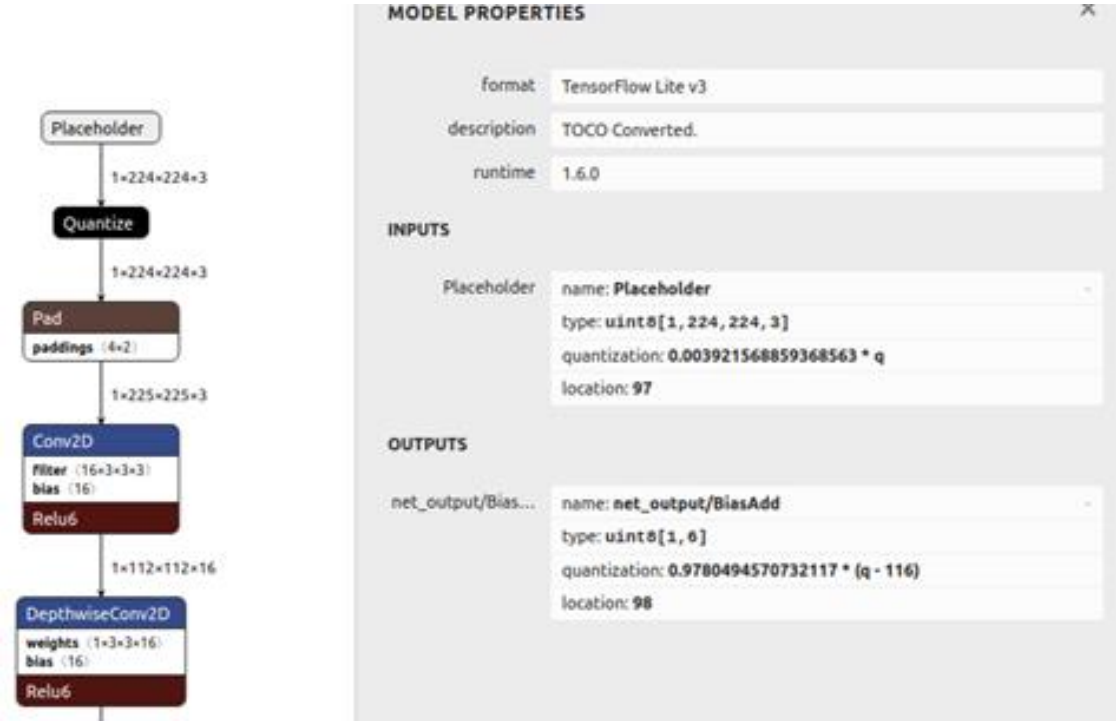

Рисунок 3.7 – Изменение точности чисел входа нейронной сети после целочисленного квантования с репрезентативным набором данных

При использовании квантования float16 в нейронную сеть добавляются блоки деквантования как для сверток, так и смещений (bias), а сами весовые коэффициенты хранятся во float16. Использование float16 позволяет увеличить скорость нейронной сети на GPU, но на CPU скорость остается прежней, так как на CPU float16 преобразуется до float32 [126]. На рисунке 3.8 показаны эти блоки и типы данных внутри них.

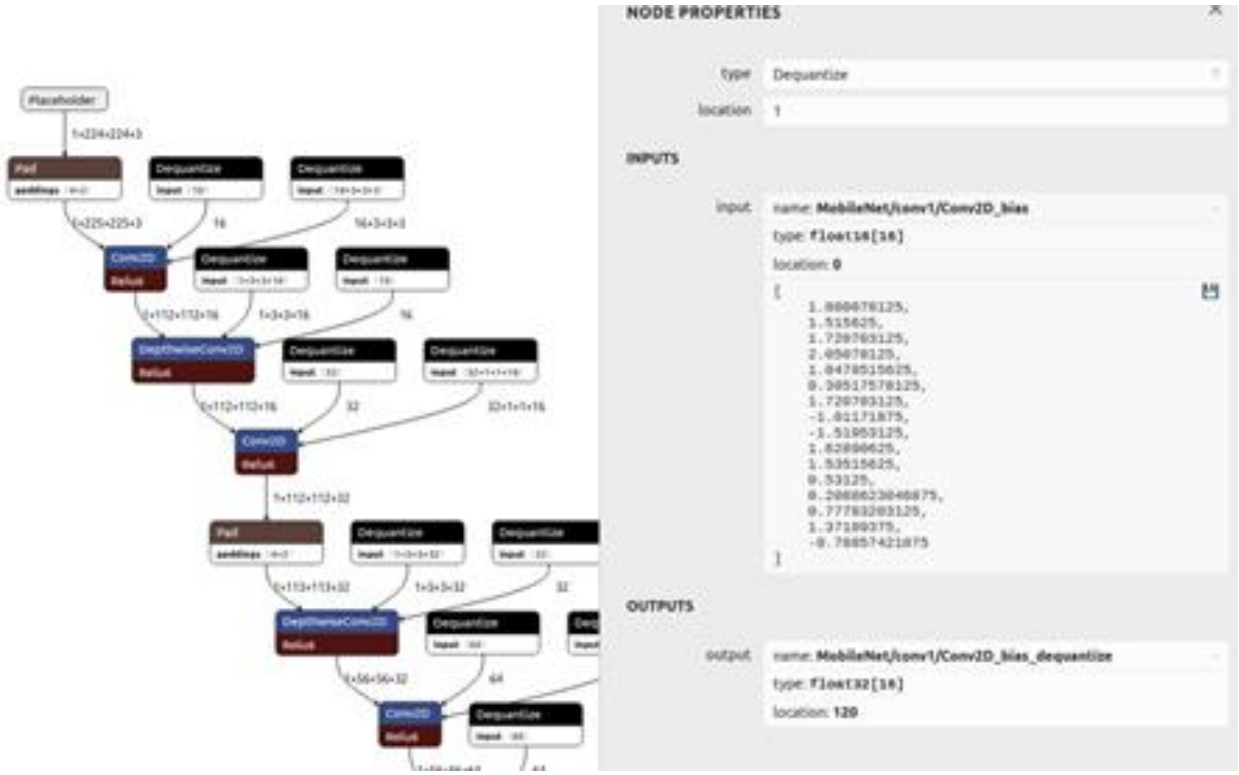

Рисунок 3.8 – Добавление блоков деквантования для весовых коэффициентов нейронной сети перед сверточными слоями после квантования во float16

Полученные после проведения квантования нейронные сети проверены на тестовой и обучающей выборках, описанных ранее. Для этих сетей (после квантования динамического диапазона, квантования с использованием репрезентативного набора данных, целочисленного квантования с использованием репрезентативного набора и квантования float16) получены характеристики: точность на тестовой выборке, точность на обучающей выборке, среднее время обработки одного изображения и размер файла модели, а также вычислены значения функции L по формуле (3.19) относительно характеристик первоначальной нейронной сети до квантования. Использование квантования влияет на различные характеристики получаемой нейронной сети. В таблице 3.1 показано сравнение исходной нейронной сети со всеми типами квантования, описанными выше.

Таблица 3.1

Характеристики нейронной сети после каждого типа квантования

	Точность на тестовой выборке, %	Точность на обучающей выборке, %	Время/изображение, мс	Размер файла, КБ	L
Исходная модель (U_0)	99,18	99,92	23	3200	0,00
Динамический диапазон (U_1)	90,26	92,12	47	834	0,21
Использование набора данных (U_2)	99,18	99,92	483	999	19,31
Использование набора данных (целочисленный вход) (U_3)	99,08	99,94	486	1000	19,44
Использование float16 (U_4)	99,18	99,92	23	1612	-0,50

Как видно из таблицы 3.1, по сравнению с исходной моделью использование квантования динамического диапазона уменьшает точность на тестовой выборке примерно на 9%, на обучающей выборке – на 7%, увеличивает время обработки изображения примерно в два раза и уменьшает размер файла модели на 74%. Использование репрезентативного набора данных оставляет точность без изменений, время обработки увеличивается на 2000% и уменьшает размер файла модели на 69%. Использование репрезентативного набора данных с целочисленным входом уменьшает точность на тестовой выборке на 0,1%, на обучающей выборке точность примерно та же, время обработки увеличивается на 2000% и уменьшает размер файла

модели на 69%. Использование float16 оставляет точность и время на обработку изображений без изменений, размер файла модели меньше на 50%.

Поэтому может быть рекомендовано квантование во float16 (U_4), при котором достигается наименьшее значение (-0,50) оптимизируемой функции L среди множества $\mathcal{U}$.

3.3. Выводы

Существующие методы поиска оптимальных гиперпараметров выполняют оптимизацию либо для нескольких критериев, либо для нескольких задач, но не объединяют оптимизацию по нескольким задачам с возможностью выбора критериев отбора гиперпараметров

Предлагаемый оригинальный метод гиперпараметрической оптимизации имеет преимущество перед существующими методами в том, что оптимизация проводится одновременно по нескольким критериям и нескольким задачам с установкой значимости критериев, обеспечивается выбор оптимальных гиперпараметров после обучения и оценки, что избавляет от необходимости повторно обучать модель, предлагаемый метод не требует обучения. При обучении нейронной сети для нескольких предприятий, имеющих обучающую выборку с отличающимися распределениями, необходимо выбрать такую сеть, которая бы давала наибольшую точность на всех этих распределениях. Путем задания критериев выбора можно управлять выбором гиперпараметров, что дает возможность искать компромисс между точностью обученной нейронной сети и временем ее обучения.

Также предлагаемый метод оптимизации вычислений нейронной сети обладает преимуществом перед имеющимися методами благодаря возможности его использования с различными архитектура нейронной сети, а также использованию с

другими математическими моделями классификациями и возможности задавать различные критерии выбора модели.

Разработанный новый метод оптимизации гиперпараметров (МТМС) при обучении нейронных сетей после моделирования на восьми устройствах дает оптимальные гиперпараметры с различными заданными критериями оптимизации по сравнению со случайным поиском быстрее примерно на 300%, с байесовской оптимизацией – на 200%.

Разработанный метод оптимизации вычислений при проведении экспериментов с различными методами квантования: динамический диапазон, Использование набора данных (целочисленный вход) и использование float16 показал, что наилучшим методам по заданным критериям точности н тестовой выборке, скорости работы нейронной сети и размера файла модели является float16. Этот метод квантования уменьшает размер файла модели в два раза без потери точности и скорости нейронной сети. Метод квантования float16 дает хорошие показатели при проверке на центральном процессоре (CPU) семейства архитектур Intel, но при использовании другого вида или архитектуры процессора наиболее оптимальным может оказаться другой вид квантования, поэтому для других процессоров рекомендуется повторное проведение эксперимента по выбору метода оптимизации вычислений.

Разработанные методы МТМС и оптимизации вычислений нейронной сети являются частью системы обучения нейронной сети в АСУТП по переработке отходов.

Глава 4. Построение устройства предварительной сортировки отходов «Сортомат» в АСУТП

4.1 Описание устройства «Сортомат»

«Сортомат» — это устройство по предварительной сортировке и приему мусора, такого как пластиковые бутылки и алюминиевые банки. Это устройство состоит из (рис. 4.1): микрокомпьютера RaspberryPi, видеокамеры, крыльчатки с поворотным механизмом и контейнера для приема мусора. В микрокомпьютере «Сортомата» размещена обученная нейронная сеть, которая обрабатывает изображения, приходящие с видеокамеры. Пользователь кладет предмет в отсек для приема, устройство принимает решение, принимать этот предмет или вернуть пользователю. Если принято решение принять предмет, то крыльчатка поворачивается для помещения предмета в контейнер и пользователю выдается чек на скидку в ближайшем магазине.

Микрокомпьютер имеет ограниченные вычислительные мощности, что ограничивает в возможностях выбора нейронной сети. Также распознавание бутылок имеет ряд особенностей по сравнению с другими задачами компьютерного зрения: бутылки отличаются друг от друга по текстуре (этикеткой и цветом бутылки), отличаются по форме (как исходной формой, так и при различных деформациях-мятостях бутылки), производители бутылок могут выпускать новые виды бутылок, а также в процессе эксплуатации «Сортомата» крыльчатка загрязняется, что сказывается на точности распознавания бутылок. Из-за этих особенностей нейронную сеть необходимо обновлять с учетом новых изображений, полученных из «Сортомата», для повышения или хотя бы для удержания точности на одном уровне.

«Сортомат» подключен к Интернету для получения новых изображений и обновления нейронной сети. Но из-за возможности расположения «Сортомата» в зоне с низким качество связи, скорость Интернета может быть низкой. По этой причине необходимо минимизировать объем передаваемых данных как с устройства, так и на устройство. Передача изображений с устройства производится в то время, когда пользователи не используют его, идеальное время для этого - ночь. Для уменьшения передаваемой в «Сортомат» нейронной сети производится уменьшение размера файла модели с помощью разработанного метода оптимизации вычислений нейронной сети. На рисунке 4.1 показаны компоненты и их взаимодействия разработанного устройства «Сортомат» (УПВ – устройство по приему вторсырья).

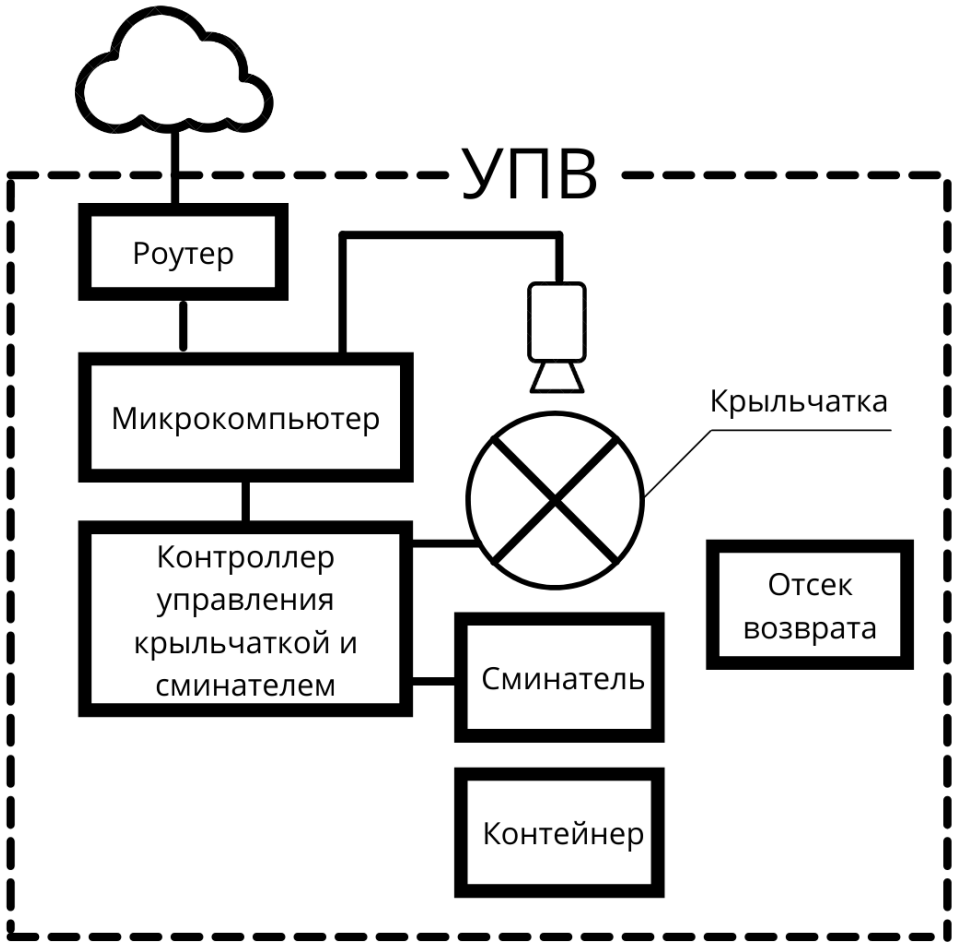

Рисунок 4.1 – Компоненты «Сортомата»

Различные устройства «Сортомат» могут иметь различные степени загрязнения крыльчатки, а пользователи могут сдавать различные бутылки, отличающиеся визуально. И для получения общей нейронной сети для всех «Сортоматов» и достигающей наибольшей точности для этих устройство применен разработанный метод гиперпараметрической оптимизации. На рисунке 4.2 показана схема обучения нескольких «Сортоматов».

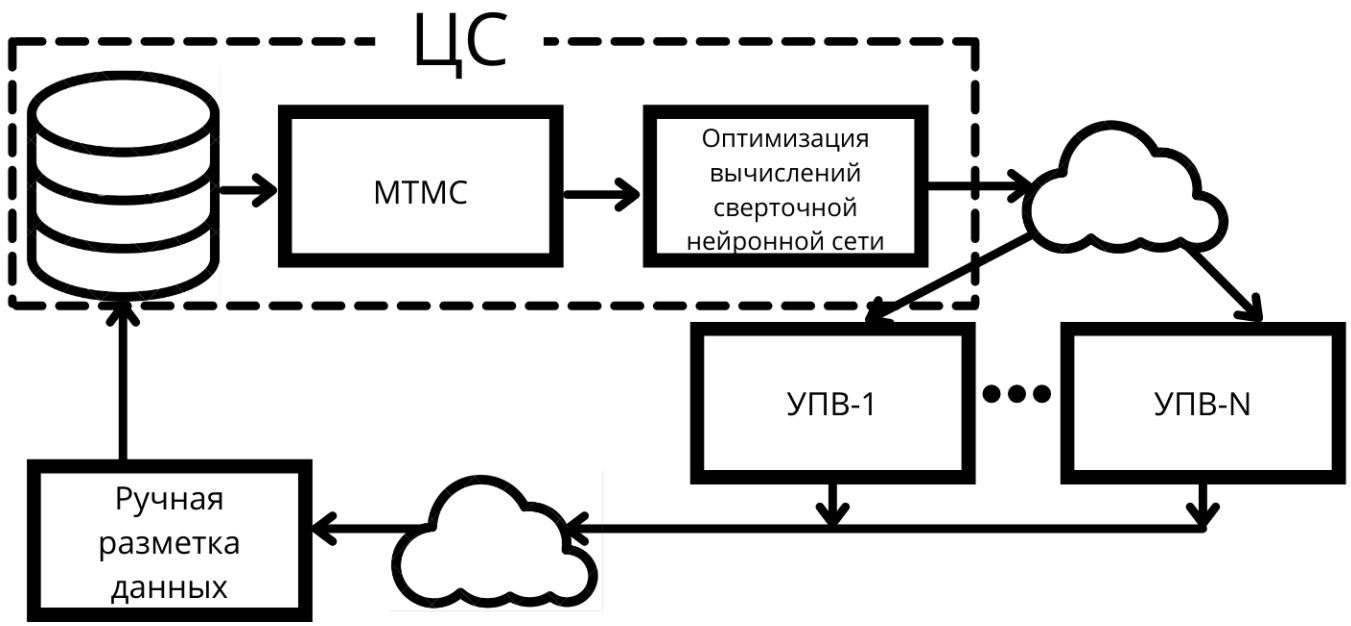

Рисунок 4.2 – Схема обучения «Сортоматов», подключенных к центральному серверу через Интернет

4.2 Обучение нейронной сети для «Сортомата» с помощью разработанных методов

Обучение нейронной сети для «Сортомата» производилось на нескольких «Сортоматов» с различными разбиениями обучающей выборки. Для этого выборка с одного «Сортомата» последовательно разбивается на три подвыборки (задачи).

Каждая подвыборка отличается загрязненностью крыльчатки, освещенностью и расположением камеры. Выбрано именно три подвыборки, так как при меньшем количестве подвыборок не удастся продемонстрировать преимущества разработанных методов оптимизации, а при большем количестве – степень загрязнения между подвыборками будет слишком слабо отличаться. На рисунке 4.3 показана схема разделения выборки на три подвыборки.

Рисунок 4.3 – Схема разделения на подвыборки

Получены изображения бутылок, банок и прочих предметов «Сортомате». В общей сложности изображения содержат семь классов предметов: ПЭТ-бутылка из-под химии, молока, масла, прозрачная ПЭТ-бутылка из-под воды, HDPE-бутылка из-под химии, алюминиевая банка и прочие предметы. Эти изображения разделены на три подвыборки. Каждая подвыборка разделена на обучающую выборку и проверочную с соотношении 80/20. Обучим нейронную сеть с выбранным методом аугментации (поворот на случайный угол от 0 до 60 градусов) и с гиперпараметрами, выбранные ранее в диссертации с помощью МТМС. Затем уменьшим размер полученной нейронной сети с помощью квантования в тип данных float16, который также был ранее выбран в этой работе. Обучение производится на облачных вычислительных мощностях на видеокарте Nvidia Tesla K80 [127]. Обучение выполнялось с помощью TensorFlow [100] и длилось 15 эпох. На рисунке 4.4 показана схема проведения эксперимента.

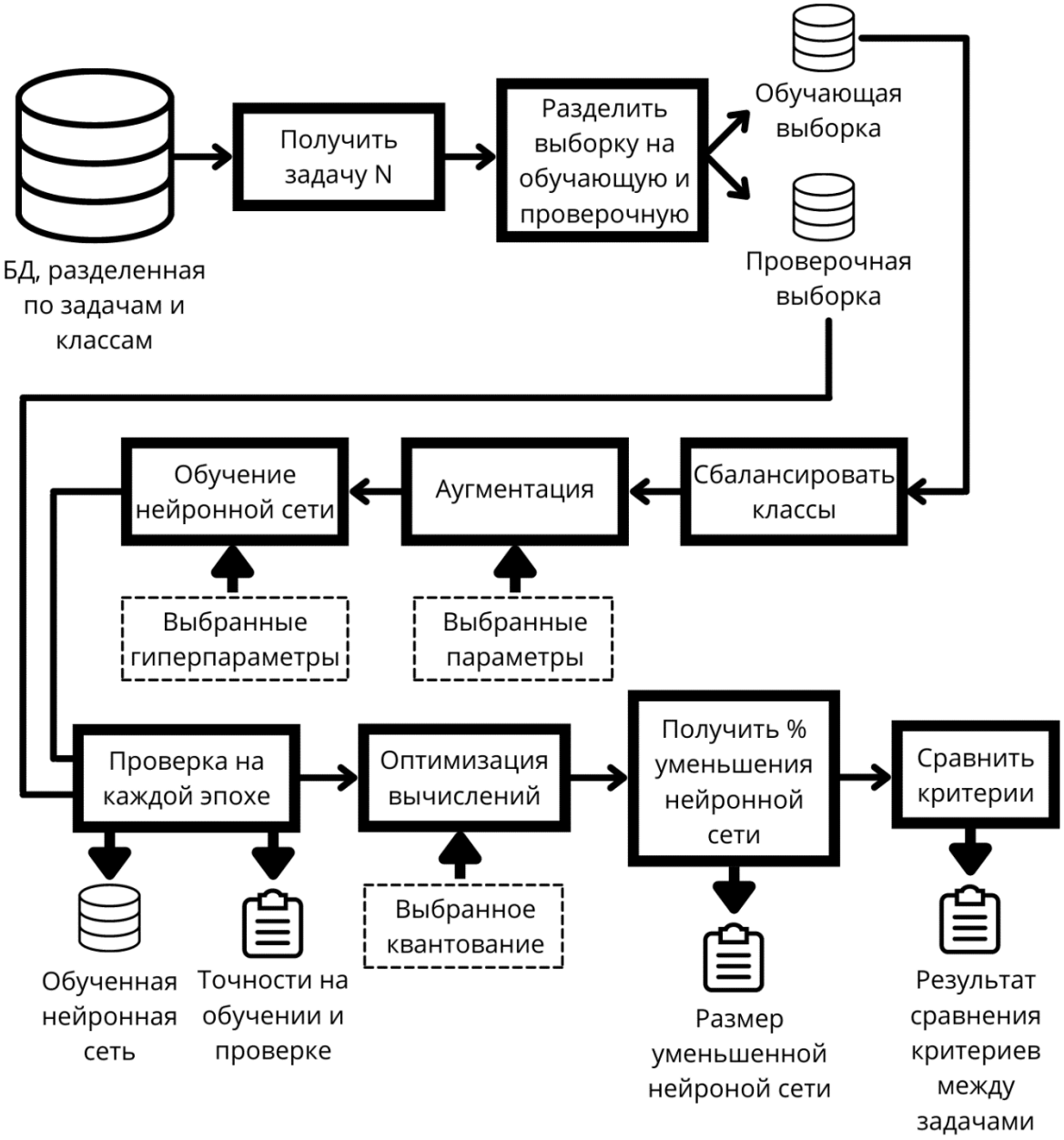

Рисунок 4.4 – Схема проведения эксперимента с разработанным методом оптимизации гиперпараметров

Процесс обучения нейронной сети и получения результатов обучения и проверки состоит из следующих этапов:

Шаг 1: извлечение указанной задачи N из базы данных (БД).

Шаг 2: перемешивание изображений каждого класса внутри указанной задачи.

Шаг 3: разделение задачи на обучающую и проверочную выборку.

Шаг 4: выполнение балансировки среди классов обучающей выборки.

Шаг 5: аугментация изображений обучающей выборки с выбранными параметрами.

Шаг 6: обучение нейронной сети с выбранными гиперпараметрами.

Шаг 7: после каждой эпохи обучения проверяем качество нейронной сети на проверочной выборке и сохраняем точности, полученные на текущей эпохи на обучающей и проверочной выборке.

Шаг 8: после завершения обучения нейронной сети вычисляем критерии (максимальная точность и скорость сходимости).

Шаг 9: производим оптимизацию вычислений (квантование) обученной нейронной сети.

Шаг 10: получаем процент уменьшения размера файла модели нейронной сети до квантования и после и сохраняем этот результат.

Шаг 11: сопоставляем результаты по каждой задаче.

В качестве критериев оптимизаций использовалась максимальная достигаемая точность и скорость сходимости обучения в соотношении 1:1, которые использовались в разработанном методе гиперпараметрической оптимизации МТМС. Эти критерии дают гиперпараметры, оптимальные как с точки зрения точности, так и с точки зрения времени обучения, с одинаковой значимостью обеих критериев. Эти оптимальные гиперпараметры представлены в Приложении Г.

Для обученной нейронной сети проведена оптимизация вычислений с помощью оригинального метода оптимизации вычислений математической модели. Для этой оптимизации использовался тип данных float16, который был ранее выбран в диссертации.

В таблице 4.1 представлены результаты проведения эксперимента на трех задачах с разработанным методом оптимизации гиперпараметров и предложенным в работе методом оптимизации вычислений.

Таблица 4.1

Результаты проведения эксперимента с разработанным методом оптимизации гиперпараметров на трех задачах классификации

	Максимальная точность на проверочной выборке, %	Номер эпохи, на которой нейронная сеть достигла сходимости	Размер исходного файла модели, КБ	Размер файла модели после квантования, КБ	На сколько уменьшился файл модели, %
Задача 1	96,61	15	12527,05	6278,27	49,88
Задача 2	96,69	15	12527,29	6278,62	49,88
Задача 3	93,08	15	12527,05	6278,27	49,88

Как видно из таблицы 4.1 нейронная сеть при использовании ранее выбранных параметров аугментации и гиперпараметров обучения на всех трех задачах достигает достаточно высокой точности классификации (более 90%), а также на всех трех задачах модель уменьшилась примерно на 50%. Обученная нейронная сеть и была использована в устройство «Сортомат».

На рисунке 4.5 показаны сопоставление гиперпараметров без их выбора (по умолчанию при запуске обучения нейронной сети) и выбранные с помощью МТМС гиперпараметры. Сопоставление производится по двух критериям: точность и номер эпохи обучения нейронной сети, на которой достигается максимальная точность.

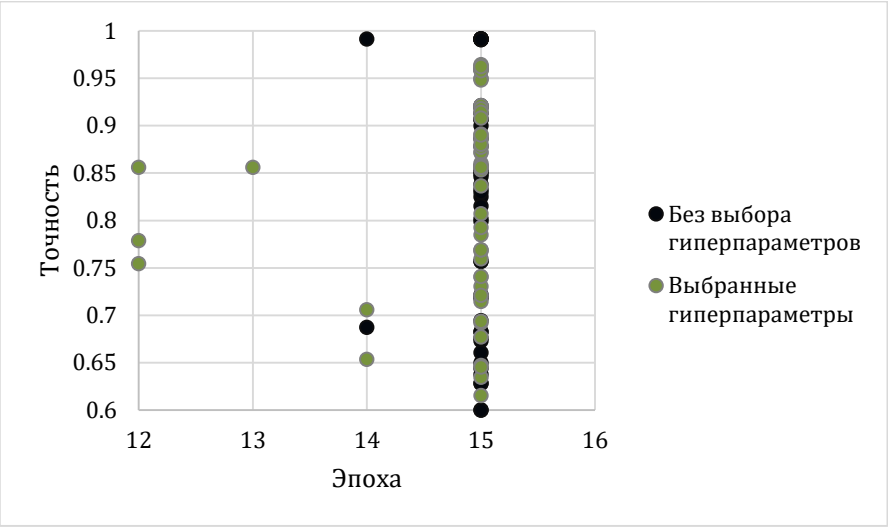

Рисунок 4.5 – Сопоставление результатов обучения нейронной сети для гиперпараметров по умолчанию и выбранных гиперпараметров

Для сравнения результатов обучения нейронной сети без выбора гиперпараметров и с выбранными гиперпараметрами проведем множественную проверку гипотез, применив поправку Бонферрони.

Первая проверяемая гипотеза – математическое ожидание точности распознавания нейронной сети, полученных без выбора гиперпараметров и с выбранными гиперпараметрами, равны Альтернативная гипотеза – математическое ожидание точности с выбранными гиперпараметрами выше, чем без выбора гиперпараметров. Проверка гипотез производится с помощью критерия Стьюдента для связанных выборок, так как нейронные сети обучались и тестировались на одних и тех же выборках. Примем уровень доверия $\alpha = 0{,}05$ и после поправки Бонферрони для двух гипотез $\alpha_1 = \alpha_2 = \frac{\alpha}{2} = 0{,}025$. Запишем проверяемую гипотезу формально:

$$\begin{cases} X_1^n = (X_{11}, \dots, X_{1n}), X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2), \\ X_2^n = (X_{21}, \dots, X_{2n}), X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2), \\ H_0: \mu_1 = \mu_2, \\ H_1: \mu_1 < \mu_2, \\ D_{1i} = X_{1i} - X_{2i}, \\ S_{d1} = \sqrt{\frac{1}{n-1}\left(\sum_i D_{1i}^2 - \frac{(\sum_i D_{1i})^2}{n}\right)}, \\ t_1 = \frac{\mathbb{E}X_1^n - \mathbb{E}X_2^n}{\frac{S_{d1}}{\sqrt{n}}}, \\ T(X_1^n, X_2^n) \sim St(n-1), \end{cases} \tag{4.1}$$

где X_1^n – точности, полученные на тестовых выборках для обученной нейронной сети без выбора гиперпараметров, X_2^n – точности, полученные для нейронной сети с выбором гиперпараметров, D_{1i} – попарные разности точностей из X_1^n и X_2^n, S_{d1} – стандартное отклонение разностей для выборок точностей.

Сделаем допущение, что выборки X_1^n и X_2^n принадлежат нормальному распределению. Размер этих выборок $n = 60$ (10 фолдов, 5 тестовых выборок и 1 обучающая выборка, на которой также проводилось тестирование). После проведенных расчетов получим следующие значения: $\mathbb{E}X_1^n = 0.7952, \mathbb{E}X_2^n = 0.8298, \mathbb{E}D_1 = -0.0352, S_{d1} = 0.0983$, статистика $t_1 = -2.7307$, уровень доверия $p_1 = 0.0044$. Так как $p_1 < \alpha_1$, то нулевая гипотеза отклоняется в пользу альтернативной, то есть точность, достигаемая с помощью выбранных гиперпараметров, выше, чем без выбора гиперпараметров.

Вторая проверяемые гипотеза – математическое ожидание эпох, на которых достигается сходимость нейронной сети, без выбора гиперпараметров и с выбранными гиперпараметрами равны. Альтернативная гипотеза – математические ожидания эпох не равны. Запишем вторую проверяемую гипотезу формально:

$$\begin{cases} X_3^n = (X_{31}, \dots, X_{3n}), X_3 \sim \mathcal{N}(\mu_3, \sigma_3^2), \\ X_4^n = (X_{41}, \dots, X_{4n}), X_4 \sim \mathcal{N}(\mu_4, \sigma_4^2), \\ H_0: \mu_3 = \mu_4, \\ H_1: \mu_3 \neq \mu_4, \\ D_{2i} = X_{3i} - X_{4i}, \\ S_{d2} = \sqrt{\frac{1}{n-1}\left(\sum_i D_{2i}^2 - \frac{(\sum_i D_{2i})^2}{n}\right)}, \\ t_2 = \frac{\mathbb{E}X_3^n - \mathbb{E}X_4^n}{\frac{S_{d2}}{\sqrt{n}}}, \\ T(X_3^n, X_4^n) \sim St(n-1), \end{cases} \tag{4.2}$$

где X_3^n – эпохи сходимости, полученные на тестовых выборках для обученной нейронной сети без выбора гиперпараметров, X_4^n – эпохи сходимости, полученные для нейронной сети с выбором гиперпараметров, D_{2i} – попарные разности точностей из X_3^n и X_4^n, S_{d2} – стандартное отклонение разностей для выборок эпох сходимостей.

После проведенных расчетов получим следующие значения: $\mathbb{E}X_3^n = 14.3833, \mathbb{E}X_4^n = 14.7000, \mathbb{E}D_2 = -0.3167, S_{d2} = 2.1680$, статистика $t_2 = -1.1314$, уровень доверия $p_2 = 0.3753$. Так как $p_2 > \alpha_2$, то нулевая гипотеза не отклоняется, то есть эпохи сходимости, полученные при обучении нейронных сетей без выбора гиперпараметров и с выбором гиперпараметров, равны.

Таким образом, точность, достигаемая с помощью выбранных гиперпараметров, выше на 3%, чем без выбора гиперпараметров, а эпоха, на которой достигается сходимость нейронной сети, примерно такая же. Вместе с применением разработанного метода увеличения точности на основе аугментации увеличение точности достигает 23%.

На рисунке 4.6 показано сопоставление результатов испытаний «Сортомата» и проведенного моделирования. Сопоставление производится также по двум критериям: точность и номер эпохи.

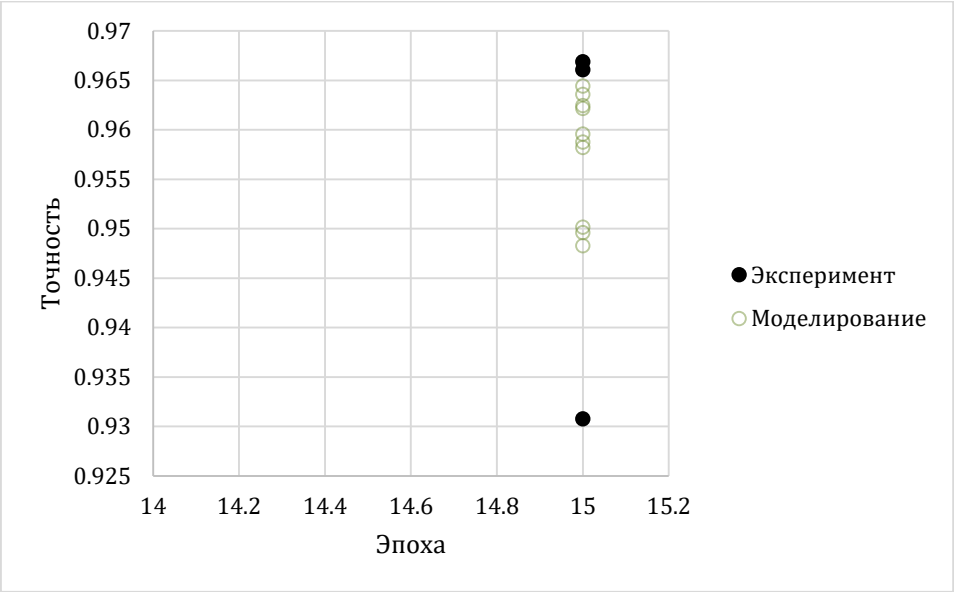

Рисунок 4.6 – Сопоставление экспериментальных результатов и результатов моделирования

Для сравнения результатов экспериментов и моделирования проведем также множественную проверку гипотез, применив поправку Бонферрони.

Первая проверяемая гипотеза – математические ожидания точностей распознавания нейронной сети, полученных в результате проведения моделирования и эксперимента, равны. Альтернативная гипотеза – математические ожидания точностей не равны. Проверка гипотез производится с помощью критерия Стьюдента для независимых выборок. Уровни доверия примем теми же: $\alpha_1 = \alpha_2 = \frac{\alpha}{2} = 0{,}025$. Запишем проверяемую гипотезу формально:

$$\begin{cases} X_5^{n_1} = \left(X_{51}, \dots, X_{5n_1}\right), X_5 \sim \mathcal{N}(\mu_5, \sigma_5^2), \\ X_6^{n_2} = \left(X_{61}, \dots, X_{6n_2}\right), X_6 \sim \mathcal{N}(\mu_6, \sigma_6^2), \\ H_0: \mu_5 = \mu_6, \\ H_1: \mu_5 \neq \mu_6, \\ t_3 = \frac{\mathbb{E}X_5^{n_1} - \mathbb{E}X_6^{n_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \\ v = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{s_1^4}{n_1^2(n_1-1)} + \frac{s_2^4}{n_2^2(n_2-1)}}, \\ T\left(X_5^{n_1}, X_6^{n_2}\right) \sim St(v), \end{cases} \tag{4.3}$$

где $X_5^{n_1}$ – точности, полученные на тестовых выборках при моделировании, $X_6^{n_2}$ – точности, полученные для нейронной сети при проведении эксперимента, v – число степеней свободы для двух независимых выборок.

Сделаем допущение, что выборки $X_5^{n_1}$ и $X_6^{n_2}$ принадлежат нормальному распределению. $n_1 = 10, n_2 = 3$. После проведенных расчетов получим следующие значения: $\mathbb{E}X_5^{n_1} = 0.9577, \mathbb{E}X_6^{n_2} = 0.9546, s_1 = 0.0003, s_2 = 0.0008$, число степеней свободы $v = 2.1939$, статистика $t_3 = 6.2497$, уровень доверия $p_3 = 0.0247$. Так как p_3 немного меньше α_1 и вычисленные математические ожидания точностей распознавания нейронной сети при поведении моделирования и эксперимента примерно равны, то нулевая гипотеза не отклоняется.

После проведенных вычислений математические ожидания эпох сходимостей равны 15, а среднеквадратичные отклонения равны 0. Поэтому можно сделать вывод, что эпохи сходимости, полученные при проведении моделирования и эксперимента, равны.

Таким образом, результаты, полученные при проведении моделирования и эксперимента, сопоставимы, расхождение точности нейронной сети не более 1% при

сравнении математических ожиданий точностей, полученных в эксперименте и в моделировании.

4.3 Внедрение полученных результатов в «Сортомат»

Внедрение обученной с помощью разработанного прототипа системы самообучения и метода оптимизации гиперпараметров проводилось на автомате по сортировке твердых бытовых отходов «Сортомат». На разработанное устройство «Сортомат» получен патент на полезную модель (Приложение Е).

На рисунке 4.7 показан внешний вид «Сортомата». Слева расположен дисплей, на котором отображается информация, что предмет принят или нет, справа вверху отсек приема, внизу – отсек возврата, если предмет распознан как прочий мусор или как предмет, не пригодный для приема.

Рисунок 4.7 – Внешний вид «Сортомата»

Для внедрения разработанного метода оптимизации гиперпараметров и вычислений нейронной сети было необходимо произвести изменения в конструкции и программном обеспечении автомата. Разработана программа склейки изображений, полученных с двух камер, для получения одного изображения, которое затем используется для предсказания класса предмета. На разработанную программу получен свидетельство (Приложение Д). Эта программа выполняет поиск ключевых точек в области пересечения двух изображений, затем определяет угол поворота и смещения изображения друг относительно друга и выполняет их склейку.

Полученные результаты подтверждают акт внедрения в «Сортомат» (Приложение Ж) и акт внедрения в учебный процесс (Приложение И).

Таким образом, внедрение системы самообучения нейронной сети в проект RVM «Sortomat» позволяет решить следующие задачи:

1. проведение автоматического обучения нейронной сети как для одного, так и для нескольких «Сортоматов» с достижением высокого качества классификации изображений;

2. дообучение нейронной сети на новых изображениях без необходимости заново проводить поиск оптимальных гиперпараметров;

3. масштабирование вычислительных мощностей с линейных увеличением скорости обучения нейронных сетей за счет высокой степени распараллеливания нагрузки на вычислительные ресурсы.

4.4 Выводы

Реализация прототипа системы самообучения нейронной сети для задачи сортировки бытовых отходов проходило на автомате по сортировке твердых бытовых отходов «Сортомат». Проведено сопоставление результатов моделирования и

экспериментов на «Сортомате». Внедрение разработанных методов в «Сортомат» показали, что достигается высокая точность (не менее 93%) классификации изображений и модель нейронной сети уменьшается примерно на 50%.

Проведено сопоставление гиперпараметров без их выбора (по умолчанию при запуске обучения нейронной сети) и выбранные с помощью МТМС гиперпараметры. Точность, достигаемая с помощью выбранных гиперпараметров, выше на 3%, чем при гиперпараметрах по умолчанию, а эпоха, на которой достигается сходимость нейронной сети, примерно такая же. Сравнение производилось по математического ожидания точности и номера эпохи, полученных для гиперпараметров по умолчанию и выбранных гиперпараметров.

Проведено сопоставление результатов испытаний «Сортомата» и моделирования. Результаты сопоставимы, расхождение точности нейронной сети не более 1% при сравнении математических ожиданий точностей, полученных в эксперименте и в моделировании.

Заключение

Основным результатом диссертационной работы является решение научной задачи, заключающейся в разработке научно-методического аппарата автоматической классификация изображений с помощью сверточных нейронных сетей в АСУТП, позволяющего уменьшить время обучения нейронной сети без потери точности. В ходе работы:

1. Проведён анализ принципов функционирования существующих систем автоматизированной сортировки бытовых отходов для выявления возможности оптимизации использования вычислительных ресурсов. В результате определено, что лучшее качество распознавания достигается с помощью сверточных нейронных сетей. Отмечено, что основными их недостатками является необходимость оптимизации гиперпараметров и обучение сети под определенную задачу распознавания.

2. Предложены метод к построению критерия оптимизации гиперпараметров для распознавания элементов бытовых отходов, метод гиперпараметрической оптимизации обучения сверточной нейронной сети с заданными критериями и метод оптимизации вычислений математической модели для минимизации вычислительных затрат на построение математической модели классификации элементов бытовых отходов, позволяющий повысить скорость получения результатов вычислений без потери их точности.

3. Предложен метод обучения сверточной нейронной сети, имеющий преимущество перед существующими методами оптимизации в возможности распараллеливать поиск оптимальных гиперпараметров с линейным увеличением скорости поиска при увеличении количества вычислительных ресурсов, что позволяет более оптимально использовать вычислительные мощности.

4. Предложены алгоритмы самообучения в автоматизированной системе сортировки бытовых отходов, которые включают поиск параметров аугментации, разработанный метод оптимизации гиперпараметров и оптимизацию вычислений обученной нейронной сети. Такая система позволяет уменьшить необходимость человеческого вмешательства в процесс обучения нейронной сети при достижении высокого качества классификации изображений на нескольких задачах распознавания (после моделирования на восьми устройствах дает оптимальные гиперпараметры с различными заданными критериями оптимизации по сравнению со случайным поиском быстрее примерно на 300%, с байесовской оптимизацией – на 200%). Применение метода увеличения точности нейронной сети дает повышение точности на 20% по сравнению с исходной нейронной сетью, а применение разработанного метода гиперпараметрической оптимизации дает увеличение точности распознавания еще на 3%. Таким образом, общее повышение точности распознавания достигает 23%.

5. Внедрение разработанных методов в «Сортомат» показали, что достигается высокая точность (не менее 93%) классификации изображений и модель нейронной сети уменьшается примерно на 50%.

Список литературы

1. Lowe D.G. Object recognition from local scale-invariant features // Proceedings of the Seventh IEEE International Conference on Computer Vision. – 1999. – Vol. 2. – P. 1150–1157.

2. McConnell R.K. Method of and apparatus for pattern recognition: pat. US4567610A USA. 1986.

3. Ojala T., Pietikäinen M., Harwood D. A comparative study of texture measures with classification based on featured distributions // Pattern Recognit. Elsevier. – 1996. – Vol. 29. – № 1. – P. 51–59.

4. Funayama R. et al. Robust interest point detector and descriptor: pat. US20090238460A1 USA. 2009.

5. LeCun Y. et al. Backpropagation applied to handwritten zip code recognition // Neural Comput. MIT Press. – 1989. – Vol. 1. – № 4. – P. 541–551.

6. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Adv. Neural Inf. Process. Syst. – 2012. – Vol. 25. – P. 1097–1105.

7. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // ArXiv Prepr. ArXiv14091556. – 2014.

8. He K. et al. Deep Residual Learning for Image Recognition // ArXiv151203385 Cs. – 2015.

9. Howard A.G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications // ArXiv Preprint ArXiv170404861. – 2017.

10. Цифровая Индустрия 4.0 | BrandVoice [Электронный ресурс] // Forbes.ru. URL: https://www.forbes.ru//brandvoice/sap/345779-chetyre-nol-v-nashu-polzu (дата обращения: 15.12.2020).

11. Что такое индустрия 4.0 и что нужно о ней знать : [Электронный ресурс] // РБК Тренды. URL: https://trends.rbc.ru/trends/industry/5e740c5b9a79470c22dd13e7 (дата обращения: 15.12.2020).

12. BMAS - Green Paper Work 4.0 [Электронный ресурс] // www.bmas.de. URL: https://www.bmas.de/EN/Services/Publications/arbeiten-4-0-greenpaper-work-4-0.html (дата обращения: 15.12.2020).

13. BMAS - White Paper Work 4.0 [Электронный ресурс] // www.bmas.de. URL: https://www.bmas.de/EN/Services/Publications/a883-white-paper.html (дата обращения: 15.12.2020).

14. Мочалова Л. А. Циркулярная экономика в контексте реализации концепции устойчивого развития //Journal of new economy. – 2020. – Т. 21. – №. 4.

14. Zeng G. et al. Continual learning of context-dependent processing in neural networks: 8 // Nat. Mach. Intell. Nature Publishing Group. – 2019. – Vol. 1. – № 8. – P. 364–372.

15. Zenke F., Poole B., Ganguli S. Continual Learning Through Synaptic Intelligence // International Conference on Machine Learning. PMLR. – 2017. – P. 3987–3995.

16. Schwarz J. et al. Progress & Compress: A scalable framework for continual learning // International Conference on Machine Learning. PMLR. – 2018. – P. 4528–4537.

17. ZenRobotics | Leader in Robotic Waste Recycling [Электронный ресурс] // ZenRobotics. URL: https://zenrobotics.com/ (дата обращения: 12.12.2020).

18. Lukka T.J. et al. ZenRobotics Recycler–Robotic sorting using machine learning // Proceedings of the International Conference on Sensor-Based Sorting (SBS) . – 2014. – P. 1–8.

19. SamurAI - Machinex Sorting Robot [Электронный ресурс] // Machinex. URL: https://www.machinexrecycling.com/products/samurai-sorting-robot/ (дата обращения: 12.12.2020).

20. AMP Neuron [Электронный ресурс] // AMP Robotics. URL: https://www.amprobotics.com/advanced-robotics (дата обращения: 12.12.2020).

21. Konečný J. et al. Federated Learning: Strategies for Improving Communication Efficiency // ArXiv161005492 Cs. – 2017.

22. Yang Q. et al. Federated Machine Learning: Concept and Applications // ACM Trans. Intell. Syst. Technol. – 2019. – Vol. 10. – № 2. – P. 12:1-12:19.

23. Bonawitz K. et al. Towards Federated Learning at Scale: System Design // ArXiv190201046 Cs Stat. – 2019.

24. Kumar S. Waste management. BoD–Books on Demand. – 2010.

25. Hoornweg D., Bhada-Tata P. What a waste: a global review of solid waste management. World Bank, Washington, DC. – 2012.

26. Kopicki R., Berg M.J., Legg L. Reuse and recycling - reverse logistics opportunities. 1993.

27. Williams E. et al. Environmental, Social, and Economic Implications of Global Reuse and Recycling of Personal Computers // Environ. Sci. Technol. American Chemical Society. – 2008. – Vol. 42. – № 17. – P. 6446–6454.

28. Sandin G., Peters G.M. Environmental impact of textile reuse and recycling – A review // J. Clean. Prod. – 2018. – Vol. 184. – P. 353–365.

30. Sabbas T. et al. Management of municipal solid waste incineration residues //Waste management. – 2003. – Vol. 23. – №. 1. – P. 61-88.

31. Diaz L. F. et al. Composting and recycling municipal solid waste. – CRC Press, 2020. ; Finstein M. S., Morris M. L. Microbiology of municipal solid waste composting //Advances in applied microbiology. – 1975. – Vol. 19. – P. 113-151.

32. Hassen A. et al. Microbial characterization during composting of municipal solid waste //Bioresource technology. – 2001. – Vol. 80. – №. 3. – P. 217-225.

33. Finstein M. S., Morris M. L. Microbiology of municipal solid waste composting //Advances in applied microbiology. – 1975. – Vol. 19. – P. 113-151.

34. Vrijheid M. Health effects of residence near hazardous waste landfill sites: a review of epidemiologic literature //Environmental health perspectives. – 2000. – Vol. 108. – №. suppl 1. – P. 101-112.

35. Ling H. I. et al. Estimation of municipal solid waste landfill settlement //Journal of geotechnical and geoenvironmental engineering. – 1998. – Vol. 124. – №. 1. – P. 21-28.

36. Oelkers E. H., Montel J. M. Phosphates and nuclear waste storage //Elements. – 2008. – Vol. 4. – №. 2. – P. 113-116.

37. Преликова Е. А., Юшин В. В., Вертакова Ю. В. Эколого-экономические приоритеты раздельного сбора отходов //Лесотехнический журнал. – 2019. – Т. 9. – №. 1 (33).

38. Байрак А. Н. Роль населения в развитии отрасли рециклирования в РФ //Вестник НГУЭУ. – 2017. – №. 1.

39. Зуева О. Н., Шахназарян С. А. Логистика возвратных потоков вторичных ресурсов //Вестник Балтийского федерального университета им. И. Канта. Серия: Гуманитарные и общественные науки. – 2014. – №. 9.

40. Гребенкин А. В., Вегнер-Козлова Е. О. Теоретические и прикладные аспекты концепции циркулярной экономики //Журнал экономической теории. – 2020. – Т. 17. – №. 2. – С. 399-411.

41. Bernath P.F. Infrared fourier transform emission spectroscopy // Chem. Soc. Rev. The Royal Society of Chemistry. – 1996. – Vol. 25. – № 2. – P. 111–115.

42. AIST:Spectral Database for Organic Compounds,SDBS [Электронный ресурс]. URL: https://sdbs.db.aist.go.jp/sdbs/cgi-bin/cre_index.cgi (дата обращения: 23.01.2021).

43. Informatics N.O. of D. and. NIST Chemistry WebBook [Электронный ресурс]. URL: https://webbook.nist.gov/chemistry/ (дата обращения: 23.01.2021).

44. Сортировка измельченного сырья | Инструмент, проверенный временем [Электронный ресурс]. URL: http://hssco.ru/sortirovka-izmelchennogo-syrya/ (дата обращения: 23.01.2021).

45. Сортировка с помощью рентген-лучей [Электронный ресурс]. URL: http://coach.refepic.ru/sortirovka-s-pomoshyu-rentgen-luchej.html (дата обращения: 15.03.2018).

46. Mortas T.N. An Automated System for Sorting Plastics by Color. – 2016.

47. Faibish S., Bacakoglu H., Goldenberg A.A. An eye-hand system for automated paper recycling // Proceedings of International Conference on Robotics and Automation. – 1997. – Vol. 1. – P. 9–14.

48. Duan F. et al. Empty bottle inspector based on machine vision // Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826). – 2004. – Vol. 6. – P. 3845–3850.

49. Huang J., Pretz T., Bian Z. Intelligent solid waste processing using optical sensor based sorting technology // 2010 3rd International Congress on Image and Signal Processing. – 2010. – Vol. 4. – P. 1657–1661.

50. Srigul W., Inrawong P., Kupimai M. Plastic classification base on correlation of RGB color // 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). – 2016. – P. 1–5.

51. Kokoulin Andrey N. et al. The Optical Method for the Plastic Waste Recognition and Sorting in a Reverse Vending Machine. – 2019. – P. 793–800.

52. Nawrocky M., Schuurman D.C., Fortuna J. Visual sorting of recyclable goods using a support vector machine // CCECE 2010. – 2010. – P. 1–4.

53. Tehrani A., Karbasi H. A novel integration of hyper-spectral imaging and neural networks to process waste electrical and electronic plastics // 2017 IEEE Conference on Technologies for Sustainability (SusTech). – 2017. – P. 1–5.

54. Sakr G.E. et al. Comparing deep learning and support vector machines for autonomous waste sorting // 2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET). – 2016. – P. 207–212.

55. Kokoulin A.N., Tur A.I., Yuzhakov A.A. Convolutional neural networks application in plastic waste recognition and sorting // 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). – 2018. – P. 1094–1098.

56. Shaikh F. et al. Waste Profiling and Analysis using Machine Learning // 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). – 2020. – P. 488–492.

57. White G. et al. WasteNet: Waste Classification at the Edge for Smart Bins // ArXiv200605873 Cs. – 2020.

58. Гудфеллоу Я., Иошуа Б., Курвилль А. Глубокое обучение // Litres. – 2018.

59. Хайкин С. Нейронные сети: полный курс, 2-е издание // Издательский дом Вильямс. – 2008.

60. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение // «Издательский дом Питер». – 2017.

61. Козлов П., Южаков А. Применение нейронной сети на основе когнитронов для распознавания образов // Нейрокомпьютеры Разработка

Применение. Закрытое акционерное общество Издательство Радиотехника. – 2014. – № 12. – С. 57–64.

62. Козлов П.В., Южаков А.А. Преобразование исходного изображения для распознавания нейронной сетью на основе неокогнитрона // Вопросы Защиты Информации. Федеральное государственное унитарное предприятие Научно-технический центр … – 2016. – № 2. – С. 32–34.

63. A Beginner’s Guide To Understanding Convolutional Neural Networks – Adit Deshpande – Engineering at Forward | UCLA CS ’19 [Электронный ресурс]. URL: https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/ (дата обращения: 23.01.2021).

64. Lin M., Chen Q., Yan S. Network in network // ArXiv Prepr. ArXiv13124400. – 2013.

65. Szegedy C. et al. Going deeper with convolutions // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – P. 1–9.

66. LeCun Y., Bengio Y., Hinton G. Deep learning: 7553 // Nature. Nature Publishing Group. – 2015. – Vol. 521. – № 7553. – P. 436–444.

67. Iandola F.N. et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size // ArXiv Preprint ArXiv160207360. – 2016.

68. Akhmetzyanov K. R., Yuzhakov A. A. Convolutional neural networks comparison for waste sorting tasks //Proceedings of Saint Petersburg Electrotechnical University Journal. – 2018. – №. 6. – С. 27-32.

69. DL4J, Torch7, Theano and Caffe [Электронный ресурс]. URL: https://deeplearning4j.org/compare-dl4j-tensorflow-pytorch (дата обращения: 23.01.2021).

70. Upton E., Halfacree G. Raspberry Pi user guide // John Wiley & Sons. – 2014.

71. UKBench Dataset [Электронный ресурс]. URL: https://archive.org/details/ukbench (дата обращения: 02.10.2018).

72. CS231n Convolutional Neural Networks for Visual Recognition [Электронный ресурс]. URL: https://cs231n.github.io/transfer-learning/ (дата обращения: 23.01.2021).

73. Deng J. et al. Imagenet: A large-scale hierarchical image database // 2009 IEEE conference on computer vision and pattern recognition. – 2009. – P. 248–255.

74. BVLC/caffe [Электронный ресурс] // GitHub. URL: https://github.com/BVLC/caffe (дата обращения: 24.01.2021).

75. MobileNet-Caffe [Электронный ресурс] // GitHub. URL: https://github.com/shicai/MobileNet-Caffe (дата обращения: 05.10.2018).

76. Epoch vs Batch Size vs Iterations | by SAGAR SHARMA | Towards Data Science [Электронный ресурс]. URL: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9 (дата обращения: 23.01.2020).

77. CS231n Convolutional Neural Networks for Visual Recognition [Электронный ресурс]. URL: https://cs231n.github.io/neural-networks-3/ (дата обращения: 23.01.2020).

78. Perez L., Wang J. The effectiveness of data augmentation in image classification using deep learning // ArXiv Prepr. ArXiv171204621. – 2017.

79. Vasconcelos C.N., Vasconcelos B.N. Convolutional neural network committees for melanoma classification with classical and expert knowledge based image transforms data augmentation // ArXiv Prepr. ArXiv170207025. – 2017.

80. Zhong Z. et al. Random erasing data augmentation // Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – Vol. 34. – № 07. – P. 13001–13008.

81. Hosseini H., Poovendran R. Semantic adversarial examples // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. – 2018. – P. 1614–1619.

82. Goodfellow I.J., Shlens J., Szegedy C. Explaining and harnessing adversarial examples // ArXiv Preprint ArXiv14126572. – 2014.

83. Moosavi-Dezfooli S.-M. et al. Universal adversarial perturbations // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2017. – P. 1765–1773.

84. Jang U., Wu X., Jha S. Objective metrics and gradient descent algorithms for adversarial examples in machine learning // Proceedings of the 33rd Annual Computer Security Applications Conference. – 2017. – P. 262–277.

85. Trusted-AI/adversarial-robustness-toolbox: Adversarial Robustness Toolbox (ART) - Python Library for Machine Learning Security - Evasion, Poisoning, Extraction, Inference [Электронный ресурс]. URL: https://github.com/Trusted-AI/adversarial-robustness-toolbox (дата обращения: 23.01.2021).

86. Bergstra J., Yamins D., Cox D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures // International conference on machine learning. – 2013. – P. 115–123.

87. Dong X. et al. Dynamical Hyperparameter Optimization via Deep Reinforcement Learning in Tracking // IEEE Trans. Pattern Anal. Mach. Intell. – 2019. – P. 1–1.

88. Mahmood A.R. et al. Benchmarking Reinforcement Learning Algorithms on Real-World Robots // Conference on Robot Learning. – 2018. – P. 561–591.

89. Tran D.-P., Nguyen G.-N., Hoang V.-D. Hyperparameter Optimization for Improving Recognition Efficiency of an Adaptive Learning System // IEEE Access. – 2020. – Vol. 8. – P. 160569–160580.

90. Dernoncourt F., Lee J.Y. Optimizing neural network hyperparameters with Gaussian processes for dialog act classification // 2016 IEEE Spoken Language Technology Workshop (SLT) . – 2016. – P. 406–413.

91. Wang L. et al. Efficient hyper-parameter optimization for NLP applications // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. – 2015. – P. 2112–2117.

92. Koriyama T., Nose T., Kobayashi T. Parametric speech synthesis based on Gaussian process regression using global variance and hyperparameter optimization // 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2014. – P. 3834–3838.

93. Bengio Y. Gradient-based optimization of hyperparameters //Neural computation. – 2000. – Vol. 12. – №. 8. – P. 1889-1900.

94. MacKay D. J. C. Comparison of approximate methods for handling hyperparameters //Neural computation. – 1999. – Vol. 11. – №. 5. – PC. 1035-1068.

95. Zhou Z., Leahy R. N., Qi J. Approximate maximum likelihood hyperparameter estimation for Gibbs priors //IEEE transactions on image processing. – 1997. – Vol. 6. – №. 6. – P. 844-861.

96. Sener O., Koltun V. Multi-task learning as multi-objective optimization // Advances in Neural Information Processing Systems. – 2018. – P. 527–538.

97. Fliege J., Svaiter B.F. Steepest descent methods for multicriteria optimization // Math. Methods Oper. Res. Springer. – 2000. – Vol. 51. – № 3. – P. 479–494.

98. Igel C. Multi-objective model selection for support vector machines // International Conference on Evolutionary Multi-Criterion Optimization. Springer. – 2005. – P. 534–546.

99. Miettinen K. Nonlinear multiobjective optimization. Springer Science & Business Media. – 2012. – Vol. 12.

100. Bergstra J.S. et al. Algorithms for hyper-parameter optimization // Advances in neural information processing systems. – 2011. – P. 2546–2554.

101. Bergstra J., Bengio Y. Random search for hyper-parameter optimization // J. Mach. Learn. Res. – 2012. – Vol. 13. – P. 281–305.

102. Snoek J., Larochelle H., Adams R.P. Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. – 2012. – P. 2951–2959.

103. Swersky K., Snoek J., Adams R.P. Multi-Task Bayesian Optimization // NIPS. – 2013. – P. 2004–2012.

104. Paria B., Kandasamy K., Póczos B. A Flexible Framework for Multi-Objective Bayesian Optimization using Random Scalarizations // PMLR. – 2020. – Vol. 115. – P. 766–776.

105. Hernández-Lobato D. et al. Predictive Entropy Search for Multi-Objective Bayesian Optimization // International Conference on Machine Learning. – 2016. – P. 1492–1501.

106. Vapnik V. Principles of risk minimization for learning theory // Advances in neural information processing systems. – 1992. – P. 831–838.

107. Akhmetzyanov K., Yuzhakov A. Waste Sorting Neural Network Architecture Optimization // 2019 International Russian Automation Conference (RusAutoCon). – 2019. – P. 1–5.

108. Learning Rate Schedules and Adaptive Learning Rate Methods for Deep Learning [Электронный ресурс] // NVIDIA. URL: https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1 (дата обращения: 10.02.2019).

109. Smith L.N. Cyclical learning rates for training neural networks // 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017. P. 464–472.

110. Stone M. Cross-validatory choice and assessment of statistical predictions // J. R. Stat. Soc. Ser. B Methodol. Wiley Online Library. – 1974. – Vol. 36. – № 2. – P. 111–133.

111. Keras [Электронный ресурс]. URL: https://github.com/keras-team/keras (дата обращения: 15.03.2019).

112. Martín Abadi, Ashish Agarwal, others. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. – 2015.

113. Bergstra J., Bengio Y. Random search for hyper-parameter optimization //Journal of machine learning research. – 2012. – Vol. 13. – №. 2.

114. Snoek J., Larochelle H., Adams R. P. Practical bayesian optimization of machine learning algorithms //arXiv preprint arXiv:1206.2944. – 2012.

115. Wistuba M., Schilling N., Schmidt-Thieme L. Hyperparameter optimization machines //2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). – IEEE, 2016. – P. 41-50.

116. Hutter F., Hoos H., Leyton-Brown K. Bayesian optimization with censored response data //arXiv preprint arXiv:1310.1947. – 2013.

117. Jacob B. et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – P. 2704–2713.

118. Han S., Mao H., Dally W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding // ArXiv Preprint ArXiv151000149. – 2015.

119. Choukroun Y. et al. Low-bit Quantization of Neural Networks for Efficient Inference. // ICCV Workshops. – 2019. – P. 3009–3018.

120. Wang K. et al. Haq: Hardware-aware automated quantization with mixed precision // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – P. 8612–8620.

121. Wu J. et al. Quantized convolutional neural networks for mobile devices // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – P. 4820–4828.

122. Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: A whitepaper // ArXiv Prepr. ArXiv180608342. – 2018.

123. TensorFlow Lite | ML для мобильных и пограничных устройств [Электронный ресурс] // TensorFlow. URL: https://www.tensorflow.org/lite?hl=ru (дата обращения: 28.01.2021).

124. Netron [Электронный ресурс]. URL: https://netron.app/ (дата обращения: 28.01.2021).

125. Целочисленное квантование после обучения | TensorFlow Lite [Электронный ресурс] // TensorFlow. URL: https://www.tensorflow.org/lite/performance/post_training_integer_quant?hl=ru (дата обращения: 28.01.2021).

126. Посттренировочное квантование float16 | TensorFlow Lite [Электронный ресурс] // TensorFlow. URL: https://www.tensorflow.org/lite/performance/post_training_float16_quant?hl=ru (дата обращения: 28.01.2021).

127. NVIDIA TESLA K80 [Электронный ресурс] // NVIDIA. URL: https://www.nvidia.com/ru-ru/data-center/tesla-k80/ (дата обращения: 28.01.2021).

Приложение А

Оценочная матрица гиперпараметров по заданным критериям

Математическое ожидание ошибки классификации	Дисперсия ошибки классификации	Математическое ожидание номера эпохи сходимости обучения	Дисперсия номера эпохи номера эпохи сходимости обучения
0,296501	0,013189	3,46	8,398
0,257044	0,021392	4,08	11,56
0,254019	0,018278	4,58	10,486
0,224982	0,017219	4,38	13,602
0,228906	0,010787	5,1	11,758
0,302321	0,016557	4,24	9,964
0,31178	0,007394	3,4	8,476
0,304317	0,021054	4,48	14,892
0,302288	0,008969	5,46	19,162
0,267959	0,01317	5,46	18,294
0,308546	0,017925	2,84	6,968
0,308067	0,005195	3,32	7,564
0,340001	0,012675	4,1	14,878
0,231876	0,006727	5,2	15,324
0,244063	0,005816	4,54	9,966
0,326111	0,015468	2,4	6,132
0,270235	0,005119	4,84	12,22
0,299199	0,011935	3,86	10,854
0,29936	0,007169	4,78	14,262
0,287972	0,007458	4,78	13,518
0,290827	0,003639	4,42	11,646
0,318165	0,007434	4,62	19,074
0,304545	0,00764	4,76	23,628
0,323974	0,004678	3,68	13,716
0,289095	0,005899	2,52	8,28
0,499966	0,017492	7,76	23,112
0,182676	0,008888	4,84	11,26
0,204553	0,01353	5,2	13,532
0,473948	0,018768	9,02	23,17
0,20963	0,009092	4,46	16,158
0,213786	0,009063	4,56	13,644
0,455531	0,016577	8,08	32,036
0,167162	0,005141	5,52	19,356

Продолжение приложения А

0.1976	0.012375	5.7	18.494
0.344638	0.005377	12.72	3.792
0.183266	0.001204	5.78	16.23
0.151462	0.000888	4.92	18.264
0.345675	0.014696	12.76	7.22
0.201308	0.004013	3.94	12.69
0.188402	0.002129	5.28	21.924
0.460398	0.010336	6.422222	33.79753
0.204407	0.009754	6.58	10.638
0.192508	0.010493	5.44	9.768
0.449016	0.017243	10.56	25.852
0.219763	0.004933	4.72	19.18
0.203108	0.009536	4.74	18.27
0.417944	0.012702	13.34	1.762
0.194197	0.008863	4.58	10.454
0.203349	0.011696	5.88	16.376
0.258641	0.014354	10.96	9.364
0.153945	0.001002	5.5	21.414
0.170482	0.001109	6	13.496
0.252831	0.022765	7.24	15.756
0.203764	0.003533	5.06	16.006
0.187454	0.001437	4.82	13.894
0.444454	0.009517	6.64	31.176
0.205746	0.008641	5.02	11.306
0.17469	0.009442	6.222222	6.528395
0.391631	0.014891	12.6	7.58
0.192518	0.012683	4.3	11.298
0.199351	0.009062	4.46	12.194
0.31494	0.009808	12.78	5.258
0.182662	0.006393	4.94	17.554
0.189565	0.011461	6.4	18.612
0.257724	0.017313	7.02	13.07
0.175609	0.001769	4.56	13.4
0.186813	0.001062	5.74	16.778
0.246542	0.02342	4.88	13.716
0.221903	0.008445	4.68	13.72
0.21908	0.002479	4.94	18.506
0.432925	0.01251	12.66667	8.790123

Окончание приложения А

0.183382	0.006932	5.266667	6.923457
0.217779	0.013007	4.94	11.742
0.295771	0.018364	11.56	8.18
0.209363	0.005249	4.16	11.96
0.195233	0.008198	4.8	16.612
0.22158	0.015157	7.533333	9.362963
0.215557	0.006073	5.08	19.868
0.190722	0.004729	5.16	22.984
0.244394	0.004988	3.96	14.72
0.158728	0.001405	6.28	16.06
0.169367	0.003136	6.36	19.764
0.203978	0.009466	6.52	20.928
0.1898	0.002221	4.6	12.404
0.195037	0.002083	5.78	19.558
0.358172	0.016738	12.86	6.682
0.172896	0.008169	5.08	13.188
0.180886	0.005564	5.711111	7.580247
0.216102	0.012196	8.36	14.336
0.209648	0.008379	4.7	11.446
0.203838	0.010983	3.84	10.228
0.244798	0.018879	4.68	15.204
0.224058	0.015037	5.02	18.882
0.176195	0.008037	5.24	14.784
0.225936	0.015765	5.12	20.184
0.175033	0.001175	5.04	12.104
0.165304	0.001235	7.42	17.862
0.20109	0.015006	6.4	21.384
0.195728	0.001913	3.74	9.738
0.198117	0.001996	4.76	14.14

Приложение Б

Оптимальные гиперпараметры для первого способа обучения

base_lr	lr_decay
0.001	0.75
0.001	0.8
0.005	0.75
0.01	0.9
0.01	0.95

Приложение В

Оптимальные гиперпараметры для второго способа обучения

base_lr	max_lr	cyclic_mode
0.0001	0.005	exp_range
0.0001	0.005	triangular2
0.0005	0.001	exp_range
0.0005	0.005	triangular2
0.001	0.0001	triangular2
0.001	0.0005	triangular
0.001	0.001	exp_range
0.001	0.005	triangular2
0.005	0.0001	triangular
0.005	0.005	triangular
0.01	0.0001	triangular
0.01	0.0001	triangular2
0.01	0.005	triangular
0.01	0.005	triangular2
0.01	0.01	triangular
0.0001	0.0001	triangular
0.0005	0.001	triangular
0.0005	0.01	triangular2
0.001	0.005	triangular
0.005	0.01	triangular

Приложение Г

Оптимальные гиперпараметры с соответствующими коэффициентами значимости критериев выбора

ϕ_0	ϕ_1	ϕ_2	ϕ_3	θ
0.5	0.5	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.0	0.5	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
1.0	0.5	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.0	0.5	0.5	base_lr=0.005, max_lr=0.0001, cyclic_mode=triangular
0.5	1.0	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	0.0	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	1.0	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	0.5	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2

Окончание приложения Г

0.5	0.5	0.5	1.0	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.0	0.0	0.5	0.5	max_lr=0.005, lr_decay=0.75
1.0	1.0	0.5	0.5	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
0.5	0.5	0.0	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.5	0.5	1.0	1.0	base_lr=0.01, max_lr=0.01, cyclic_mode=triangular
1.0	0.0	0.0	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.0	1.0	0.0	0.0	base_lr=0.0001, max_lr=0.005, cyclic_mode=triangular2
0.0	0.0	1.0	0.0	max_lr=0.005, lr_decay=0.75
0.0	0.0	0.0	1.0	base_lr=0.0005, max_lr=0.001, cyclic_mode=exp_range

Приложение Д

Свидетельство о государственной регистрации программы для ЭВМ «Программа бинокулярного зрения с учетом расстояния до объекта»

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2019664159

Программа бинокулярного зрения с учетом расстояния до объекта

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (RU)***

Авторы: ***Ахметзянов Кирилл Раисович (RU), Тур Александр Игоревич (RU), Кокоулин Андрей Николаевич (RU), Южаков Александр Анатольевич (RU)***

Заявка № **2019662351**
Дата поступления **08 октября 2019 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***30 октября 2019 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Приложение Е

Патент на полезную модель

«АВТОМАТ ПО ПРИЁМУ ТАРЫ»

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ПАТЕНТ

НА ПОЛЕЗНУЮ МОДЕЛЬ

№ 188755

АВТОМАТ ПО ПРИЁМУ ТАРЫ

Патентообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего образования "Пермский национальный исследовательский политехнический университет" (RU)*

Авторы: *Тур Александр Игоревич (RU), Кокоулин Андрей Николаевич (RU), Ахметзянов Кирилл Раисович (RU), Южаков Александр Анатольевич (RU)*

Заявка № **2018145675**
Приоритет полезной модели **21 декабря 2018 г.**
Дата государственной регистрации в Государственном реестре полезных моделей Российской Федерации **23 апреля 2019 г.**
Срок действия исключительного права на полезную модель истекает **21 декабря 2028 г.**

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Приложение Ж

Акт внедрения в «Сортомат»

Общество с ограниченной ответственностью
«ГОРНЫЕ ВЕНТИЛЯЦИОННЫЕ УСТРОЙСТВА»

614013, Пермь, ул. Академика Королева, 21, оф. 213

АКТ

о внедрении результатов кандидатской диссертационной работы
Ахметзянова Кирилла Раисовича

Комиссия в составе:
Председатель комиссии: Николаев А.В., генеральный директор ООО «Горные вентиляционные устройства», канд. тех. наук.
Член комиссии: Кокоулин А.Н., специалист технической разработки, канд.тех.наук, Полыгалов С.В., специалист отдела проектирования и разработки, канд. тех. наук.
составила настоящий акт о том, что результаты диссертационной работы «НЕЙРОСЕТЕВЫЕ МЕТОДЫ И АЛГОРИТМЫ САМООБУЧЕНИЯ ПРИ ОБРАБОТКЕ ДАННЫХ В СИСТЕМЕ АВТОМАТИЗАЦИИ ПРОЦЕССА СОРТИРОВКИ БЫТОВЫХ ОТХОДОВ» использованы при проектировании и реализации системы оптического распознавания в аппарате по приему тары «Сортомат».

При создании модификации «Сортомат 1.0»:

1. Был использован и внедрен процесс обучения нейронной сети в соответствии с предложенным автором методом гиперпараметрической оптимизации, оптимизации вычислений математической модели и самообучения нейронной сети. При этом:

- реализован метод, выполняющий настройку гиперпараметров нейронной сети для распознавания бытовых отходов с учетом критериев скорости и точности распознавания с использованием многозадачной оптимизации; что обеспечило повышение эффективности используемых вычислительных ресурсов сервера обучения нейронной сети для распознавания объекта на изображении на 30% по сравнению с предыдущими показателями,

- реализован метод оптимизации вычислений, выполняющий поиск преобразования весовых коэффициентов для повышения скорости работы нейронной сети и сокращения времени распознавания бытовых отходов без потери точности (увеличена скорость обучения нейронной сети на 10%).

2. Разработанные в диссертации модели (аналитическая и имитационная) были применены для расчета характеристик процесса распознавания бытовых отходов (скорость и точность распознавания).

3. Повышение эффективности использования вычислительных мощностей и повышения автоматизации процесса обучения нейронной сети уменьшено необходимое человеческое воздействие на процесс обучения нейронной сети, что повысило скорость подсистемы распознавания бытовых отходов на 15%.

Кроме того, предложенные Ахметзяновым К.Р. в его диссертационной работе решения будут использованы при создании новой подсистемы распознавания бытовых отходов «Сортомат».

Председатель комиссии _____ А.В. Николаев

Члены комиссии _____ А.Н. Кокоулин
_____ С.В. Полыгалов

Приложение И

Акт внедрения в учебный процесс

УТВЕРЖДАЮ

Проректор по учебной работе
Пермского национального
исследовательского политехнического
университета

_____ / Н.В. Лобов /

«14» сентября 2021 г.

АКТ

о внедрении результатов

полученных Ахметзяновым Кириллом Равилевичем

при выполнении диссертационной работы

на соискание ученой степени кандидата технических наук

«Нейро-сетевые методы и алгоритмы самообучения при обработке данных в системе автоматизации процесса сортировки бытовых отходов»

Комиссия в составе:

Председатель: Фрейман Владимир Исаакович, доктор технических наук, доцент, профессор, заместитель заведующего кафедрой «Автоматика и телемеханика» по учебной и методической работе

Члены комиссии: Гончаровский Олег Владленович, кандидат технических наук, доцент, доцент кафедры «Автоматика и телемеханика»,

Тур Александр Игоревич, кандидат технических наук, доцент кафедры «Автоматика и телемеханика»,

составила настоящий акт о том, что результаты диссертационной работы «Нейро-сетевые методы и алгоритмы самообучения при обработке данных в системе автоматизации процесса сортировки бытовых отходов» соискателя Ахметзянова К.Р. используются для проведения лекционных и практических занятий, лабораторных работ, курсового проектирования в рамках программы магистратуры по направлению подготовки 15.04.06 «Мехатроника и робототехника», магистерская программа «Автономные сервисные роботы».

Предложенные научные основы создания и исследования принципов обучения нейронных сетей с учетом многокритериальной и многозадачной оптимизации нашли применение:

Окончание приложения И

- в дисциплине «Методы идентификации зрительных объектов в робототехнике» программы магистратуры «Автономные сервисные роботы». Результаты диссертационного исследования применены в рамках лекционных материалов и лабораторных практикумов с целью демонстрации возможностей оптимизации систем визуального распознавания на примере многокритериальной многозадачной гиперпараметрической оптимизации и оптимизации вычислений математической модели, позволяющей повысить точность системы;
- в дисциплине «Разработка систем распознавания образов для автономных сервисных роботов» программы магистратуры «Автономные сервисные роботы». Результаты диссертационного исследования применены в рамках лабораторных практикумов в составе заданий по созданию оптических подсистем распознавания объектов и ориентирования в пространстве, основанных на применении самообучений нейронной сетей в условиях ограниченных вычислительных мощностей платформы.

Эффект от внедрения результатов диссертационной работы заключается в повышении уровня освоения профессиональных компетенций и их компонентов (знаний, умений и владений) в области проектирования элементов систем управления автономными сервисными роботами. Это соответствует требованиям Федеральных государственных образовательных стандартов высшего образования нового поколения, построенных с учетом требований профессиональных стандартов.

Председатель комиссии:

доктор технических наук, доцент / Фрейман В.И. /

Члены комиссии:

кандидат технических наук, доцент / Гончаровский О.В. /

кандидат технических наук, доцент / Тур А.И. /

«14» сентября 2021 г.