

Пермский национальный исследовательский политехнический университет

На правах рукописи

Тур Александр Игоревич

**ИЕРАРХИЧЕСКИЙ МЕТОД РАСПОЗНАВАНИЯ В ПОДСИСТЕМАХ  
МАШИННОГО ЗРЕНИЯ АСУТП СОРТИРОВКИ И УТИЛИЗАЦИИ  
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 – 2020

## Оглавление

Введение.....	5
Глава 1. Анализ средств и методов обработки и распознавания растровых изображений в АСУТП.....	12
1.1 Применение модулей технического зрения в АСУТП.....	12
1.2 Анализ методов обработки растровых изображений.....	19
1.3 Анализ инструментальных средств обработки растровых изображений .....	25
1.4 Анализ ресурсных возможностей систем машинного зрения в контексте Big Data .....	28
1.5 Выводы по главе.....	32
Глава 2. Иерархический метод обработки растровых изображений в автоматизированных системах управления с применением модулей машинного зрения.....	34
2.1 Сравнение иерархического и других подходов в АСУТП .....	34
2.2 Алгоритм иерархической обработки растровых изображений.....	37
2.3 Применение иерархического подхода к существующим проектам .....	42
2.3.1 Распознавание статических объектов на изображениях.....	42
2.3.2 Распознавание динамических объектов на изображениях .....	50
2.4 Выводы по главе.....	52
Глава 3. Моделирование процесса иерархического хранения и обработки растровых изображений.....	54
3.1 Описание условий моделирования процесса .....	54
3.2 Аналитическая модель процесса иерархической обработки растровых изображений .....	56

3.2.1 Описание процесса создания аналитической модели .....	56
3.2.2 Расчёт основных показателей аналитической модели.....	59
3.3 Имитационная модель процесса иерархической обработки растровых изображений .....	61
3.3.1 Описание процесса создания имитационной модели .....	61
3.3.2 Расчёт основных показателей имитационной модели .....	63
3.4 Анализ результатов аналитического и имитационного моделирования.....	65
3.6 Выводы по главе.....	68
4. Внедрение иерархического метода распознавания в АСУТП по сортировке мусора.....	70
4.1 Реализация прототипа системы иерархического распознавания растровых изображений .....	70
4.2 Тестирование системы иерархического хранения и обработки растровых изображений .....	72
4.3 Внедрение результатов в проекте RVM «Sortomat».....	77
4.4 Выводы по главе.....	79
Заключение .....	81
Список сокращений .....	83
Список терминов .....	84
Список литературы .....	90
Приложение А .....	105
Приложение Б.....	109
Приложение В.....	110
Приложение Г .....	111
Приложение Д.....	122

Приложение Е.....	123
Приложение Ж.....	125

## Введение

### **Актуальность проблемы и степень разработанности темы.**

Подсистемы технического зрения в АСУТП позволяют выполнять задачи без вреда и опасности для здоровья человека. Такие задачи в большом количестве присутствуют на вредных и тяжёлых производствах. Одно из таких – утилизация бытовых отходов. Первым этапом данного производства является сортировка. На этом этапе входная фракция очищается от предметов, не попадающих под обрабатываемый тип. Наиболее производительными методами сортировки являются те, которые используют промышленные линии, состоящие из транспортной ленты, устройство анализа объекта и сортирующего механизма. На рынке сортировки и утилизации бытовых отходов широко представлены автоматы таких компаний как «Хэнань Хунцзи», «Европлан Системс», «АРК» «ВторТех». Лидеры рынка этих машин применяют в качестве устройства анализа – спектрометры. Однако такие линии являются дорогими в покупке и обслуживании. Более экономичным вариантом является использование в качестве устройства анализа оптической системы сортировки, применяющей машинное зрение для распознавания объектов.

Функция компьютерного зрения в автоматизированных системах является одной из основных и уже достаточно хорошо изучена. Значительный вклад в создание алгоритмов распознавания объектов внесли такие зарубежные учёные, как Альфред Хаар, Герберт Фриман, Крис Харрис, Майк Стефенс, Навнит Далал, Билл Триггс, Ян Лекун, Яньцинг Цзя, а также отечественные – Ю.Б. Зубарев, В.П. Дворкович, А.В. Дворкович.

Известны методы, алгоритмы и инструменты распознавания [4, 20, 62, 70, 71, 88, 117], обладающие определёнными преимуществами. Однако почти все из них базируются на полном (или близком к нему) переборе пикселей всего изображения. Ярким примером такого подхода являются современные свёрточные нейронные сети (LeNet, AlexNet и др.). Анализ больших изображений таким подходом занимает достаточно много времени. Кроме того, чем мощнее

метод распознавания тем, больше ему требуется вычислительных ресурсов. Для свёрточных нейронных сетей это означает увеличение количества памяти выделяемой для распознавания объекта и обработки промежуточных результатов работы процесса. Поэтому в настоящее время актуальна задача ресурсоёмкости процесса распознавания.

Стационарные исследовательские системы обладают достаточной вычислительной мощностью и временем для распознавания объекта с требуемой точностью, а производственные и мобильные комплексы (направленные на автоматизацию технологических процессов) испытывают серьезные ресурсные ограничения, что в итоге негативно сказывается на качестве распознавания. [31, 96, 98, 99, 100, 104, 120] Особенно актуальна эта проблема для робототехнических мобильных комплексов требующих высокой надежности и отказоустойчивости, т.к. задачи повышения надежности и снижения затрачиваемых вычислительных ресурсов находятся в противоречии. По существу, появляется оптимизационная задача с критериями точности, скорости (производительности), аппаратурных затрат. [36, 60, 73, 96, 99, 100, 104, 120]

Самые простые методы распознавания изображения базируется на определении контура и ключевых точек объекта на изображении (например, основанные на цепном коде Фримена, детекторе Хариса). Их преимуществом является малое время выполнения распознавания и небольшой объём требуемых вычислительных ресурсов. Недостатком является высокие требования к входным данным и невозможность распознать объект при малейшем изменении его вида. Более продвинутые методы распознавания (например, основанные на HOG, SURF, SIFT) ощутимо медленнее, однако позволяют распознавать объекты имеющие отличие во внешнем виде от оригинала (поворот относительно плоскости изображения и даже незначительное изменение формы). Кроме того, для корректной работы данные методы требуют больше вычислительных ресурсов. Ещё больше вычислительных ресурсов требует применение методов, основанных на использовании свёрточных нейронных сетей. Однако при этом они показывают наилучшие результаты распознавания объектов (имеют большую

устойчивость к помехам мешающим проводить распознавание), а время выполнения процесса незначительно больше, чем у описанных выше.

В диссертационной работе исследуется задача распознавания объектов, заданных многоканальными изображениями, в условиях ограничения требований к вычислительным ресурсам. Для её реализации предлагается использовать поэтапное уточнение местоположения распознаваемого объекта на изображении – иерархический метод распознавания объекта на изображении. Такой подход позволит анализировать изображение не целиком, а по частям, что снижает время распознавания и количество требуемых вычислительных ресурсов. Это означает, что возможно применение методов распознавания, основанных на использовании свёрточных нейронных сетей, при более низких требованиях к вычислительной платформе без снижения качества распознавания.

**Объект исследования:** системы машинного зрения, применяемые в АСУТП для обеспечения обработки данных, представленных в графическом виде.

**Предмет исследования:** методы обработки больших данных, представленных в графическом виде в составе автоматизированной системы с ограниченными вычислительными ресурсами.

**Цель работы:** реализация эффективных по вычислительной сложности и качеству распознавания процессов обработки данных, представленных в графическом виде в АСУТП в условиях ограниченных вычислительных ресурсов.

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

1. Провести анализ существующих систем распознавания информации, представленной в графическом виде, с целью выявления возможности повышения эффективности использования вычислительных ресурсов;

2. Разработать оригинальный иерархический метод обработки данных, представленных в графическом виде, с целью снижения ресурсозатратности и повышения скорости работы систем при сохранении требуемой достоверности распознавания;

3. Построить аналитическую и имитационную модели иерархической системы распознавания информации, представленной в графическом виде в условиях ограниченных вычислительных ресурсов;

4. Провести апробацию и внедрение разрабатываемой иерархической системы распознавания.

**Методы исследований** основаны на теории СМО, теории нейронных сетей, теории телетрафика, системном и статистическом анализе, на аналитическом и имитационном моделировании, теории планирования и обработки результатов эксперимента.

**Основные положения, выносимые на защиту:**

1. Оригинальный иерархический метод обработки информации, представленной в графическом виде, в АСУТП сортировки и утилизации бытовых отходов на основе иерархического подхода, позволяющий снижать требования к аппаратной базе системы без уменьшения достоверности результатов распознавания (*п. 8. Формализованные методы анализа, синтеза, исследования и оптимизация модульных структур систем сбора и обработки данных в АСУТП, АСУП, АСТПП и др.*);

2. Аналитическая и имитационная модели иерархической системы распознавания информации, представленной в графическом виде, в условиях ограниченных вычислительных ресурсов, обеспечивающие возможность оценки эффективности подсистемы распознавания АСУТП (*п. 15. Теоретические основы, методы и алгоритмы интеллектуализации решения прикладных задач при построении АСУ широкого назначения (АСУТП, АСУП, АСТПП и др.)*).

3. Программно-аппаратурная реализация системы иерархического метода распознавания информации, представленной в графическом виде в условиях ограниченных вычислительных ресурсов в АСУТП сортировки и утилизации бытовых отходов.

**Научная новизна:**

1. Предложен и разработан оригинальный иерархический метод обработки информации, представленной в графическом виде, в

автоматизированной системе управления на основе иерархического подхода, отличающийся тем, что реализует поэтапное уточнение области поиска объекта на изображении при сохранении уровня достоверности результатов распознавания, что обеспечивает снижение требований к ресурсоемкости системы;

2. Разработаны специализированные (для описанной задачи) аналитическая и имитационная модели иерархической системы распознавания информации, представленной в графическом виде, особенностью которых является учет перераспределения ограниченных вычислительных ресурсов, что позволяет исследовать применимость иерархического метода распознавания к конкретной системе, абстрагируясь от сложных нелинейных расчётов и трудоёмких натурных экспериментов.

**Теоретическая значимость** заключается в:

- создание иерархического метода распознавания, основанного на поэтапном уточнении местоположения распознаваемого объекта на изображении;
- разработке моделей иерархической системы распознавания объекта на изображении, в условиях ограниченных вычислительных ресурсов, учитывающей возможность перераспределения освободившихся вычислительных ресурсов.

**Практическая значимость** заключается в разработке и программной реализации метода иерархического распознавания объекта на изображении, применимого для большинства алгоритмов распознавания, позволяющего:

- уменьшить общее время распознавания объекта на изображении в условиях ограниченных вычислительных ресурсов, при сохранении уровня достоверности получаемых результатов;
- снизить количество требуемых вычислительных ресурсов для распознавания объекта на отдельно взятом изображении, при сохранении уровня достоверности получаемых результатов.

Это позволяет использовать в системах с ресурсными ограничениями (например, производственные и мобильные комплексы АСУТП) ресурсоёмкие алгоритмы распознавания объектов без потери качества распознавания.

**Достоверность и обоснованность результатов.** Общие тенденции, полученные в результате исследования, не противоречат результатам, представленными в литературе другими исследователями, а также подтверждаются сопоставлением результатов аналитического и имитационного моделирования с экспериментальными данными, полученными на физической вычислительной платформе, и результатами внедрения.

**Апробация работы.** Основные результаты диссертационной работы представлялись и обсуждались на IX международной Интернет-конференции молодых ученых, аспирантов и студентов «Инновационные технологии: теория, инструменты, практика» InnoTech 2017 (Россия, г. Пермь, 2017 г.), XIV всероссийской школе-конференции молодых ученых «Управление большими системами» УБС'2017 (Россия, г. Пермь, 2017 г.), 12 международной научно-технической конференции ПСТИ-2017 (Россия, г. Суздаль, 2017 г.), XXI международной конференции по мягким вычислениям и измерениям (Россия, г. Санкт-Петербург, 2018 г.), 2018 и 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (Россия, г. Санкт-Петербург, 2018 г. и 2019 г.), First international geographical conference of north asian countries "China-Mongolia-Russia economic corridor: geographical and environmental factors and territorial development opportunities" (IOP Conference Series: Earth and Environmental Science) (Россия, г. Иркутск, 2019 г.), 19th International Multidisciplinary Scientific GeoConference SGEM-2019 (Болгария, г. Албена, 2019), IV всероссийской научно-практической конференции «Искусственный интеллект в решении актуальных социальных и экономических проблем XXI века» (Россия, г. Пермь, 2019 г.).

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

**Публикации.** Основные результаты диссертации изложены в шестнадцати статьях, шесть из которых — в журналах ВАК, в том числе один патент на полезную модель и одно свидетельство на программу для ЭВМ, шесть — в

изданиях, входящих в международные базы цитирования (Scopus и Web of Science), остальные – в тезисах докладов, материалах конференций и прочих источниках.

**Объем и структура работы.** Диссертация состоит из введения, четырёх глав и заключения, списка литературы из 129 наименований и 7 приложений. Полный объем диссертации составляет 131 страниц, из которых 81 страницы занимает основной текст диссертации, включающий 31 рисунка и 5 таблиц.

# **Глава 1. Анализ средств и методов обработки и распознавания растровых изображений в АСУТП**

## **1.1 Применение модулей технического зрения в АСУТП**

Машинное зрение – это научное направление в области искусственного интеллекта, в частности робототехники, и связанные с ним технологии получения изображений объектов реального мира, их обработки и использования полученных данных для решения разного рода прикладных задач без участия (полного или частичного) человека. Система машинного зрения использует одну или несколько видеокамер, устройство аналого-цифрового преобразования (АЦП) и цифровой обработки сигналов (ЦОС). Полученные данные поступают в компьютер или робота. [90, 123]

Подсистемы технического зрения в АСУТП позволяют выполнять задачи без вреда и опасности для здоровья человека. Такие задачи в большом количестве присутствуют на вредных и тяжёлых производствах. Одно из таких – утилизация бытовых отходов. АСУТП именно такого производства будет рассмотрена в этой работе.

Мусор в России стал серьёзной проблемой, требующей решения на всех уровнях власти, а также подключения населения и предприятий страны. Всё, что связано с данным вопросом, регламентирует 458 ФЗ «Об отходах производства и потребления». Он был принят 29 декабря 2014 года и установил новые требования по переработке для производителей и импортеров. Основные цели закона заключались в том, чтобы побудить компании максимизировать объемы товаров, которые подвергаются переработке и повторному использованию, и минимизировать объем общих отходов.

До него на территории России действовал другой законодательный акт – ФЗ-89, вышедший ещё в 1998 году. Так как экологическая ситуация стремительно

ухудшается, а отходов становится всё больше, законодательный акт быстро устарел. [118] Было принято решение пересмотреть его и на основе старого документа создать новый. Прежний законодательный акт (ФЗ-89) имел отношение только к отходам промышленного сектора. Новая же версия рассматривает также и потребительский тип. Также в неё включили и вопросы перевозки и переработки всех видов мусора.

Новый закон рассматривает все этапы передвижения мусора от потребителя к полигонам и перерабатывающим предприятиям. В нём сформированы правила транспортирования, условия сбора и т. д. Так например все организации обязуются:

- самостоятельно утилизировать и/или перерабатывать собственные отходы;
- если нет возможности следовать предыдущему пункту, заключать соглашения с утилизирующими и/или перерабатывающими компаниями, имеющими лицензию на совершение такой деятельности;
- платить экологический сбор, если не выполняется ни один из предыдущих пунктов.

Данный закон постоянно корректируется – выходят новые редакции. Закон постепенно приближается к международным нормам защите окружающей среды. Особо актуальна данная проблема стала на стыке 2019 и 2020 года. Все организации стремятся приобрести статус «зелёных» и начинают развивать различные проекты, связанные со сбором и переработкой мусора.

Под термином утилизация понимают использование вторсырья для дальнейшего его рециклинга. Рециклинг – это непосредственно сам процесс переработки отходов в материал, который можно использовать вторично. В результате этого действия получают продукцию – из макулатуры туалетную бумагу, из пластика — другие пластиковые изделия, из битого стекла новые изделия (т.е. в процессе утилизации обеспечивается вторичное использование или переработка отходов, которые не могут быть использованы более в исходном виде). [84, 89, 95, 102, 103, 106, 115] Уничтожение не является синонимом

утилизации. Как правило, уничтожение является последним звеном в данном процессе.

Под переработкой понимают деятельность, которая направлена на превращение отходов во вторичное сырье (которое в дальнейшем можно будет использовать для создания других изделий), либо энергетические ресурсы. В процессе переработки, сырье может подвергаться обработке, в ходе которой меняется его физическое, биологическое или химическое состояние. Делается это с целью его использования в дальнейшем. Обрабатывают многие виды пластмасс, бумагу, стекло и т.д. Переработку некоторых материалов невыгодно осуществлять, так как для этого затрачивается слишком много ресурсов, поэтому такие материалы подвергаются уничтожению. [108-110]

Однако у процесса переработки и у процесса утилизаций первым этапом является сортировка. [65] На этом этапе входная фракция очищается от предметов, не попадающих под обрабатываемый тип – из общей массы отдельно извлекают бумажные изделия, пластиковые, стеклянные и т.д. Сортировка может производиться разными способами и в разных объёмах – это могут быть баки отдельного сбора мусора (в которые сами жители населённого пункта выбрасывают уже отсортированные отходы), пункты приёма отходов (в которых наёмные рабочие сортируют принимаемый у жителей мусор, как показана на рисунке 1.1).

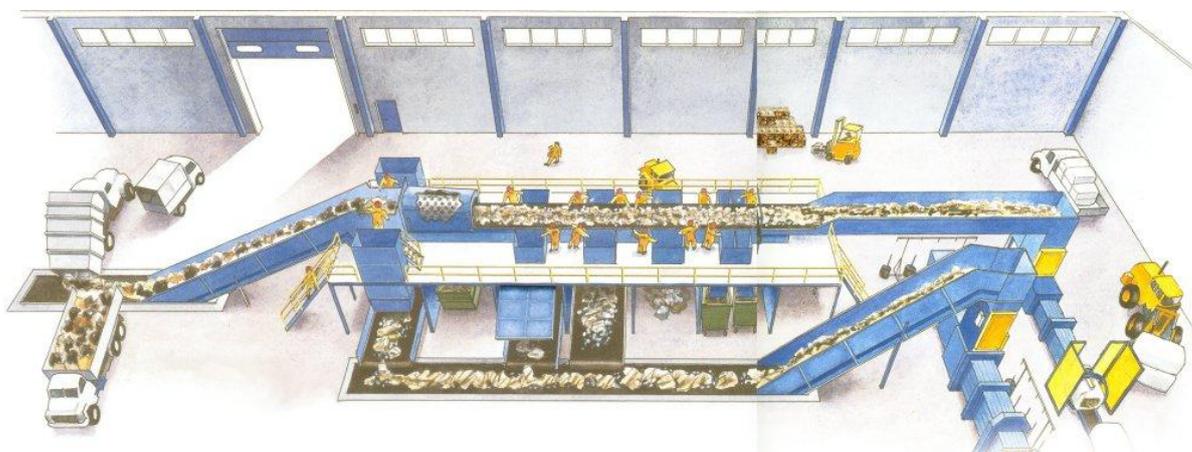


Рисунок 1.1 – Пример линии ручной сортировки ТКО

В перечисленных случаях сортировка производится по кодам переработки. Коды переработки (рис. 1.2) – специальные знаки, применяются для обозначения материала, из которого изготовлен предмет, и упрощения процедуры сортировки перед его отправкой на переработку для вторичного использования. Такие знаки обычно ставят на батарейках, аккумуляторах, изделиях из стекла, металла, бумаги, пластмассы, изделий из органических материалов природного происхождения: древесины, пробки, джутового волокна, хлопка.



Рисунок 1.2 – Код переработки утверждённый Европейской комиссией 97/129/ЕС, обозначающий применение в изделии полиэтилентерефталата

Чтобы распознать материал, необходимо установить соответствие его внешнего вида с таблицей используемых символов (приложение А). Так например, если внешний вид знака представляет собой квадрат, в котором расположен треугольник из объемных стрелок, внутри символа набор из арабских цифр, под символом находятся английские буквы то перед Вами код переработки утверждённый Европейской комиссией 97/129/ЕС, обозначающие материал изделия. Именно этот код переработки на упаковках и учитывается при сортировке ТКО.

Согласно рекомендациям разработчиков, наносимые на упаковку коды должны быть достаточно крупными, так как её материал недостаточно ценный,

чтобы тратить время на чтение мелкого кода. Однако производитель не всегда следует этим рекомендациям – уменьшает код (для сохранения красоты упаковки) или располагает его в незаметном месте (применительно к ПЭТ бутылкам – доньшко). Такое разнообразие осложняет сортировку. Кроме того, часто упаковка бывает повреждённой и код, как таковой, может отсутствовать на ней полностью. В таких случаях сортировщик руководствуется внешним видом упаковки (бутылки из под напитков производят из полиэтилентерефталат, бутылки из-под бытовой химии – полиэтилена высокой плотности и так далее). Совокупность этих факторов значительно ограничивает скорость выполнения сортировки, а также позволяет допускать ошибки, которые негативно сказываются на качестве вторичного сырья. [113, 114]

Более производительными являются методы сортировки, использующие промышленные линии (которые, как правило, находятся на самих перерабатывающих и утилизирующих предприятиях и сортируют мусор, привозимый компаниями, занимающимися вывозом мусора из населённого пункта). [27] Большая часть таких сортировочных конвейеров представляют собой транспортную ленту (по которой движется мусор), устройство анализа объекта и сортирующего механизма. Лидеры рынка этих машин применяют в качестве устройства анализа – спектрометры. [6, 17, 42] Пример такой линии представлен на рисунке 1.3.

Спектрометр – оптический прибор, предназначенный для фиксации отражённых волн от исследуемого объекта (спектра), их количественной обработки и дальнейшего анализа. Для получения анализируемого спектра используется определенный вид излучения (рентгеновское, лазерное, искровое), а его регистрация происходит путем флуоресценции. Как правило, в ходе исследования измеряются интенсивность излучения, его длина, волна, частота, но могут быть определены и другие параметры. Приборы работают в диапазоне длин волн: от гамма до инфракрасного излучения (для каждого типа отходов применяют свой диапазон). Они позволяют почти безошибочно (анализ многослойных объектов часто даёт неправильный результат) и быстро

распознавать тип материала. Однако данные приборы достаточно дорогие из-за сложных в производстве специальных оптических элементов.



Рисунок 1.3 – Пример промышленной линии сортировки ТКО, распознающий отходы с помощью спектрометра

Альтернативой могут быть системы машинного зрения. Их производительность немного ниже (в первую очередь зависит от характеристик камеры и общей скорости обработки информации), определяется не сам материал, а его свойства. Самые простые системы могут сортировать исходя из прозрачности материала. [53] Пример такой системы представлен на рисунке 1.4. Однако такой способ уместен только для специфичных задач и неуниверсален для большинства случаев. Более сложные линии могут распознавать форму объекта, подразумевающую использование искомого материала (система узнаёт ПЭТ пластик по форме продукции из этого материала – бутылки, контейнеры и прочее). Но всё ещё остаётся вероятность неправильной сортировки в случае слишком плотного (разные образцы расположены слишком близко друг к другу) потока отходов и в случае повреждения формы объекта (помятые или порванные бутылки). Однако, одновременно с этими недостатками система получается в разы дешевле и проще в обслуживании, чем линия применяющая спектрометры для распознавания. [28]



Рисунок 1.4 – Пример промышленной линии сортировки ТКО, распознающий отходы с оптического датчика по прозрачности материала

Данный вариант системы актуален для линий, где сортируется неизмельчённый мусор, и нет требований к высоким скоростям сортировки. Именно такой тип системы (система с применением оптических методов сортировки ТКО) будет рассмотрен в данной работе.

В обычных условиях задача визуального наблюдения и определения с технической точки зрения довольно простая. Однако в случае АСУТП достаточно часто возникают проблемы, связанные с необходимостью использования контроллеров со значительным ограничением по вычислительной мощности. [33, 92, 112] В таком случае процесс обработки изображения может быть медленным и не удовлетворять требованиям к скорости производства. Другим вариантом осложняющим работу таких систем может быть желание обрабатывать большое количество изображений одновременно [70, 100] или большие по объёму изображения. [88, 96, 99, 104, 120] Однако проблема остаётся той же – снижение скорости обработки изображения из-за нехватки вычислительных ресурсов.

## 1.2 Анализ методов обработки растровых изображений

Задачу реализации машинного зрения, в общем случае, можно разделить на три этапа работы с изображением:

1. фильтрация изображения для вычленения информации;
2. логическая обработка результатов фильтрации;
3. принятие решения на основе логической обработки. [107, 128]

В зависимости от конкретных условий реализаций и поставленной задачи некоторые этапы могут объединяться или исключаться полностью. Основными критериями оценки работы системы машинного зрения являются:

- качество распознавания (чаще всего выражается в проценте ошибок при распознавании);
- быстродействие;
- общая сложность реализации. [87, 128]

Под фильтрацией изображения обычно подразумеваются все действия, направленные на извлечение информации из изображения. Логическая обработка результатов – применение различных алгоритмов извлечения характерных особенностей изображения, при сравнении которых (на этапе принятия решения) и выносятся вердикт о схожести изображений. Наиболее важным этапом в данной цепочке является применение алгоритмов извлечения характерных особенностей изображения – от него зависит набор подготовительных действий с изображением и возможности при определении схожести изображений.

Группа самых простых способов сравнения изображений – расчёт площади объекта на изображении. Эту задачу можно решать двумя разными способами – получением размеров объекта и дальнейшим расчётом по формулам, либо, если объект сложной формы, подсчётом количества пикселей занимаемых объектом и сопоставлением их с предварительно рассчитанной площадью. Второй способ прост и достаточно быстрый. Однако может иметь погрешности при неправильных настройках и некачественных входных данных. Для первого же

способа требуются дополнительные вычисления, что замедляет общую работу алгоритма. Однако оба способа могут быть применимы только в узком спектре задач. [87] Группа этих алгоритмов не подходит для задач, реализуемых АСУТП утилизации бытовых отходов, прежде всего из-за большого количества форм и размеров объектов (банок, бутылок и прочих контейнеров содержащих пластик), а также из-за нестабильности их внешнего вида из-за деформаций и различных повреждений в ходе эксплуатации. Кроме того, даже частичное перекрытие другим предметом негативно отразится на результатах распознавания, что значительно усложняет процесс функционирования сортировочного конвейера.

Более сложные способы – вычисление контура объекта. Он оптимален при распознавании объектов на контрастном фоне или заранее приведённых к бинарному виду. На изображении проводится выявление контура объекта любым алгоритмом контурного анализа. Примером может быть цепной код Фримана. [116] Существует группа алгоритмов, оценивающих изображения по отдельным особым участкам изображения – особым точкам. Они также применяются для распознавания объектов и их условно можно разделить на три класса. [129]

Первый класс (самый быстрый, менее сложный) – особые точки, являющиеся стабильными на протяжении секунд. Такие точки служат для того, чтобы вести объект между соседними кадрами видео, или для сведения изображения с соседних камер. К таким точкам можно отнести локальные максимумы изображения, углы на изображении (детектор Хариса), точки в которых достигается максимумы дисперсии, определённые градиенты и.т.д. [129]

Второй класс (ощутимо медленнее) – особые точки, являющиеся стабильными при смене освещения и небольших движениях объекта. Такие точки служат в первую очередь для обучения и последующей классификации типов объектов. К таким точкам относятся точки, найденные методом гистограмм направленных градиентов (HOG). [129]

Третий класс (самый медленный из перечисленных, но показывающий самые продуктивные результаты) – стабильные точки. Самые распространённые методы из этой группы – SURF и SIFT. Они позволяют находить особые точки даже при

повороте изображения. Расчёт таких точек осуществляется дольше по сравнению с остальными методами, но за достаточно ограниченное время. [129]

Данные три класса направлены на обнаружение и сопоставление в точности совпадающих объектов, но уже способны узнавать объекты иного размера (однотипные по дизайну бутылки разной ёмкости). Однако проблема того, что объект на изображении может иметь небольшие отличия от оригинала (например, нестабильности их внешнего вида из-за деформаций и различных повреждений в ходе эксплуатации, или из-за частичного перекрытия другим предметом) всё ещё не позволяет применять их. Для подобных задач применяются более сложные поисковые алгоритмы, использующие нейронные сети

Свёрточная нейронная сеть (CNN – convolutional neural network) — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения (deep learning). Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв (convolution layers) и субдискретизирующих слоёв (subsampling layers или pooling layers, слоёв подвыборки). Структура сети – однонаправленная (без обратных связей), принципиально многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов (передаточная функция) – любая, по выбору исследователя. [25, 37, 71]

Название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения (рисунок 1.5).

Работа свёрточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям, и далее к ещё более абстрактным деталям вплоть до выделения понятий высокого уровня. При этом сеть самонастраивается и вырабатывает необходимую иерархию абстрактных признаков (последовательности карт признаков), фильтруя маловажные детали и выделяя существенное. [5, 7, 30, 64]

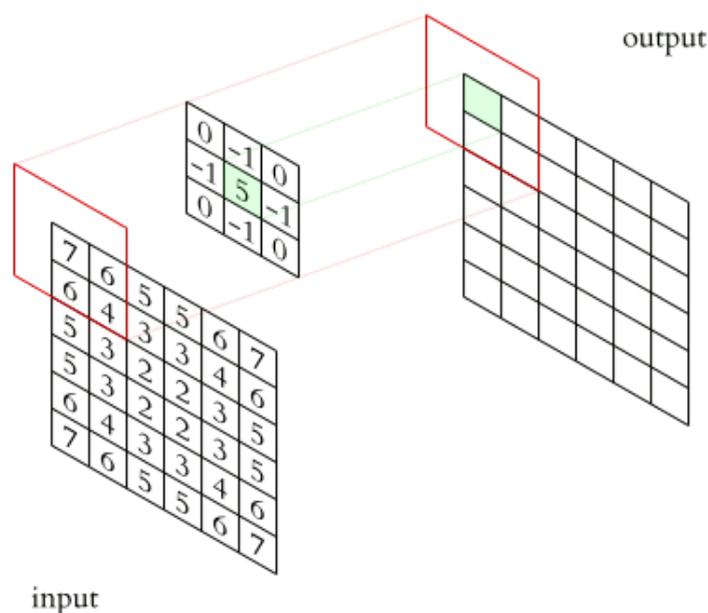


Рисунок 1.5 – иллюстрация работы функции свёртки

В отличие от остальных методов машинного обучения, требующих предварительного извлечения информативных признаков для осуществления классификации, сверточные сети решают обе эти задачи в процессе обучения, напрямую используя цветные каналы изображений. CNN является многослойной сетью прямого распространения, отличительная особенность которой состоит в наличии групп нейронов, имеющих общие весовые коэффициенты. Такие группы нейронов образуют двумерные плоскости. Другими словами, сверточные слои для входного изображения вычисляют карты признаков путем его свертки с небольшими ядрами, коэффициенты которых настраиваются в процессе обучения нейронной сети. Так как сеть является многослойной, то сверточные слои способны извлекать высокоуровневые особенности, используя признаки, полученные на предыдущих слоях. После сверточного слоя выполняется нелинейное преобразование карт, для чего наиболее часто применяют активационные функции типа ReLU и tanh [24]. Также используется операция объединения (pooling), которая сокращает размеры карт, что позволяет уменьшить влияние шума и обеспечивает некоторую инвариантность к масштабу классифицируемых объектов. Для сокращения размерностей карт признаков обычно используют операции взятия максимума или усреднения значений в

окрестностях пикселей [52], а также случайный выбор значений [75]. Последние несколько слоев CNN представляют собой полносвязную нейронную сеть, осуществляющую непосредственную классификацию выделенных признаков.

Сверточные нейронные сети демонстрируют выдающееся качество в решении задачи распознавания, но главным препятствием их применения в системах является скорость работы, которой недостаточно для обработки видеопотоков высокого разрешения в режиме реального времени.

Для уменьшения времени распознавания объектов на больших изображениях часто применяют так называемый «Coarse-to-Fine» (от грубого к точному) подход.

Грубый и точный поиск изображений – это общая методика повышения эффективности любого алгоритма поиска. Грубый поиск использует версию изображения в низком разрешении, обычно получаемую путем построения пирамиды изображений. Поскольку эта версия меньше оригинальной, количество обрабатываемых пикселей значительно меньше, и, соответственно, поиск выполняется быстрее. Чтобы повысить достоверность обнаружения и уточнить оценки местоположения, процесс поиска повторяется с более высоким разрешением (более «точный» масштаб), но только в тех областях, представляющих интерес (англ. «Region Of Interest», ROI), которые были идентифицированы с более низким разрешением и могут содержать один из объектов, заданных в поиске – точный поиск. Таким образом, значительно уменьшается фактическая область поиска. Этот процесс повторяется при последовательно более высоких разрешениях до тех пор, пока не будет достигнута достаточная достоверность и точность или не будет произведен поиск исходного изображения. [59]

Этот метод хорошо масштабируется в соответствии с размером изображения и может быстро обрабатывать большой объём данных, используя относительно простые, настраиваемые вручную компоненты распознавания образов. Нейронные сети предоставляют дополнительный набор методов, так как они могут изучать сложные шаблоны, но они плохо масштабируются с размером изображения. [12, 59]

Новым преимуществом объединения «Coarse-to-Fine» подхода и нейронных сетей является возможность использовать контекстную информацию, под которой подразумеваются любые видимые характеристики объекта, которые могут частично или полностью совпадать с характеристиками искомого объекта. Контекст может помочь сузить область поиска в низких разрешениях и улучшить производительность обнаружения, поскольку это, по определению, актуальная информация. Каждая сеть получает некоторые функции, извлеченные из поискового скользящего окна в изображении, и его вывод интерпретируется, как вероятность найти один из искомых объектов в центр окна. [128]

При оценке движения чаще всего используются два представления изображения – традиционная пиксельная сетка и представления слоя движения. Первый из них прост, но страдает от более высокой вероятности неоднозначного сопоставления, в то время как основная проблема использования последнего заключается в трудностях правильной сегментации слоя и параметризации движения слоя. Такие представления могут уменьшить вычислительную сложность по сравнению с представлениями на основе пикселей из-за гораздо меньшего числа областей. Также, учитывая меньший размер областей, вероятность того, что ошибки цветовой сегментации распространяются в процессе сопоставления, уменьшается по сравнению с представлением на основе слоев. [11-15, 40, 41, 78]

На сегодняшний день наиболее распространённым методом обработки растровых изображений является применение различных нейронных сетей (тип нейронной сети зависит от конкретной задачи, но преимущественно используются свёрточные нейронные сети). Поэтому в данной диссертационной работе будет рассмотрено именно их применение, но с точки зрения повышения их скорости работы и ресурсоёмкости посредством иерархического подхода (Coarse-to-Fine).

Для реализации описанного подхода необходимо применение программных инструментов (программ, библиотек, сред программирования), позволяющих решать задачи распознавания информации представленной в графическом виде.

### 1.3 Анализ инструментальных средств обработки растровых изображений

OpenCV (Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Может свободно использоваться в академических и коммерческих целях – распространяется в условиях лицензии BSD. Для предлагаемых задач данная библиотека предоставляет возможность использования достаточно простого и в тоже время эффективного алгоритма основанного на каскадах Хаара. Ключевой особенностью признаков Хаара является наибольшая, по сравнению с остальными признаками, скорость. При использовании интегрального представления изображения, признаки Хаара могут вычисляться за постоянное время (примерно 60 процессорных инструкций на признак из двух областей). [50, 92, 123, 128]

TensorFlow – открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия. Применяется как для исследований, так и для разработки собственных продуктов Google. Основной API для работы с библиотекой реализован для большинства современных языков программирования. [25, 61, 91, 111]

Caffe – среда для глубинного обучения, разработанная Яньцинем Цзя (Yangqing Jia) в процессе подготовки своей диссертации в университете Беркли. Caffe является открытым программным обеспечением, распространяемым под лицензией BSD license. Название Caffe произошло от сокращения «Convolution Architecture For Feature Extraction» (Свёрточная архитектура для распознавания признаков). [9, 19, 29]

На базе Tensorflow и Caffe наиболее известными и популярными являются нейронные сети AlexNet и LeNet. LeNet – сверточная нейронная сеть, представленная ещё в 1998 году французским исследователем Яном Лекуном (Yann LeCun), как развитие модели неокогнитрон (neocognitron). Её базовая модель состоит из трёх типов слоёв: свёрточные (convolutional) слои, субдискретизирующие (subsampling, подвыборка) слои и слои "обычной" нейронной сети – перцептрона (рисунок 1.6).

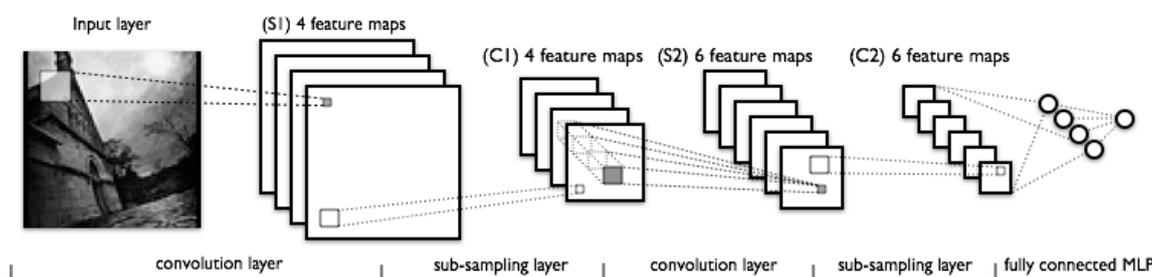


Рисунок 1.6 – Слои сети LeNet

AlexNet — сверточная нейронная сеть, которая оказала большое влияние на развитие машинного обучения, в особенности — на алгоритмы компьютерного зрения. Сеть с большим отрывом выиграла конкурс по распознаванию изображений ImageNet LSVRC-2012 в 2012 году (с количеством ошибок 15,3% против 26,2% у второго места). С точки зрения топологии сети это почти тот же LeNet, просто увеличенный в тысячу раз. Добавились еще несколько сверточных слоев, а размер ядер свертки уменьшается от входа сети к выходу (рисунок 1.7). Это объясняется тем, что вначале пиксели сильно скоррелированы, и рецепторную область можно смело брать большую, мы все равно теряем мало информации. Также были внедрены другие возможности позволяющие предотвратить переобучение: Dropout (RIP), Data Augmentation и ReLu. [22, 23, 26, 66, 68, 69, 77, 125]

Как показывает практика, CNN имеют значительное преимущество над классическими алгоритмами (например, каскады Хаара), так как имеют более сложный аппарат формирования признаков распознаваемого объекта. Это

позволяет детектировать объект даже в сложных условиях, препятствующих распознаванию. Однако имеют большую ресурсозатратность [55], что вынуждает применять более оптимальные системы хранения информации.

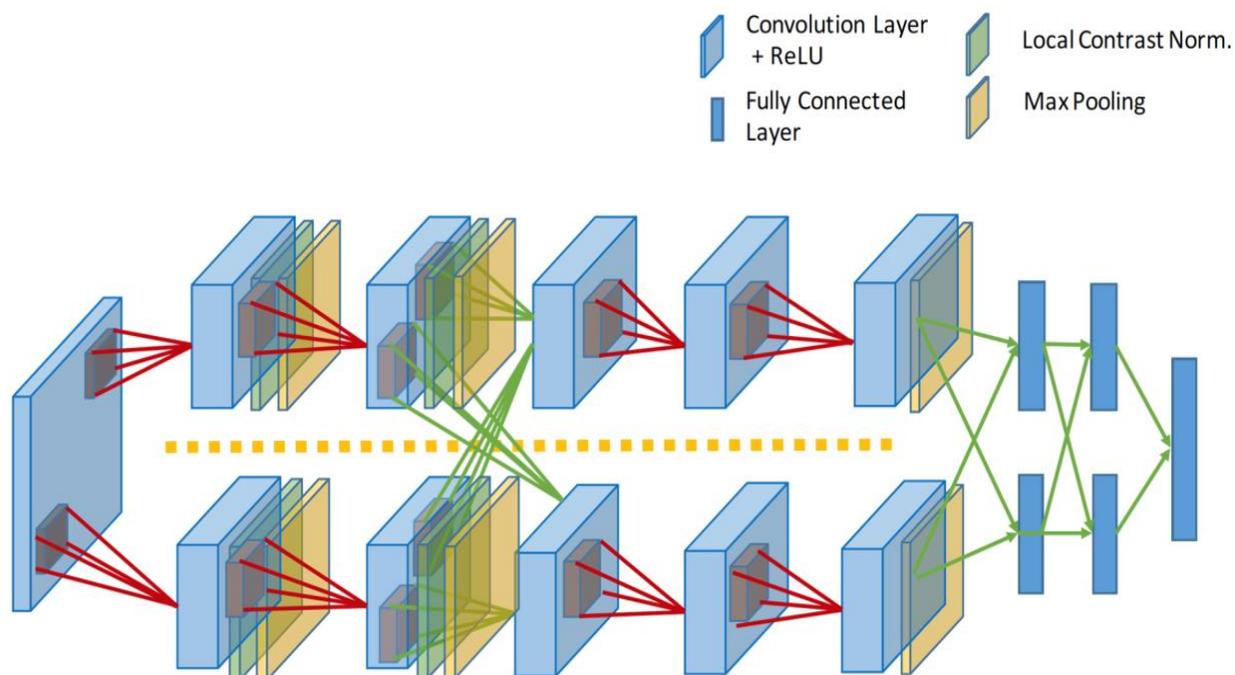


Рисунок 1.7 – Слои сети AlexNet

Проблема сортировки отходов требует от разработчиков достаточно высокого качества распознавания объекта. Поэтому применение более быстрых и менее ресурсозатратных алгоритмов редко бывает оправданным из-за их низкой точности. Из-за экономии в данном аспекте конечная система может оказаться сложнее и дороже, чем система с применением более сложных алгоритмов. [128] Опираясь на данное утверждение, в данной работе будет рассматриваться процесс распознавания объектов при помощи нейронных сетей. Это, в свою очередь, обязывает рассмотреть вопрос ресурсозатратности таких систем.

## 1.4 Анализ ресурсных возможностей систем машинного зрения в контексте Big Data

На сегодняшний день во всех областях человеческой деятельности часто приходится сталкиваться с необходимостью обработки большого объёма данных. Данная проблема получила название «Big data» или «Большие данные».

Большие данные, как правило, определяются в статьях тремя буквами V – «Volume», «Variety» и «Velocity» («Объём», «Разнообразие» и «Скорость» соответственно). [49] «Volume» обозначает количество данных. Оно, безусловно, может стать проблемой в том случае, когда большие объёмы данных обрабатываются при помощи технологий напрямую не предназначенных для этого. «Variety» – сложность данных (получение из нескольких источников, различные структуры представления). Данный показатель может значительно усложнить процесс обработки (дополнительные преобразование и формирование разных структур для разных задач), что негативно отразится на его эффективности. «Velocity» подразумевает под собой скорость доступа к данным.

Первые два V часто обусловлены техническими процессами и не могут быть изменены:

- как правило, нет возможности уменьшить объём данных из-за статистических и качественных требований;
- снижение количества источников во многих принципиально не возможно.

Кроме того, даже если предполагаемая система обработки больших данных избежала обозначенных выше проблем, то пороговое значение её эффективности будет зависеть от скорости чтения/записи информации. Первым шагом к разрешению данного вопроса является максимальное сокращение этапов перемещения информации при её обработке – в идеальном случае большие данные обрабатываются непосредственно на том же компьютере, где и хранятся. Кроме того, данный шаг позволит сократить количество возможных ошибок при передаче информации. [101]

Вторым шагом является выбор средств хранения информации. Стоит отметить, что такие устройства должны быть достаточного объёма для хранения тех самых больших данных. Этот вопрос разрешается различными типами систем хранения данных. Кроме того немаловажна и пропускная способность такой системы. Дешёвые варианты с низкой пропускной способностью серьёзно ограничат быстродействие порогом своей максимальной скорости. Варианты с хорошей пропускной способностью не будут ограничивать обработку, но из-за высокой стоимости не всегда могут быть экономически эффективны. [43. 105] Обеспечение высокой производительности ввода-вывода позволит сохранять промежуточные и окончательные результаты вычисления в адекватный период времени и предоставлять своевременный результат. Кроме того, минимизация времени ввода-вывода также необходима для уменьшения вероятности возникновения ошибки во время этой критической транзакции и для минимизации времени на сохранение контрольных точек. [43. 105]

Одним из решений, которое позволит минимизировать время ввода-вывода для алгоритмов и обеспечить надлежащую пропускную способность является так называемый «Burst Buffer». Под этим термином понимается включение промежуточного уровня хранения, представляющего собой устройства хранения с высокой пропускной способностью и сравнительно небольшим объемом. [38, 67, 122] По своей сути они действуют как участок подготовки необходимых данных или как кэш с обратной записью для высокопроизводительных систем хранения. Исследование, проведенное с использованием суперкомпьютера Intrepid однозначно демонстрирует, что использование «Burst Buffer» является эффективным решением как для повышения производительности стадии ввода-вывода, так и для сокращения нагрузки на внешнюю систему хранения. [10, 38, 67]

Таким образом, дополнительным преимуществом решения «Burst Buffer» является значительное сокращение расходов на организацию внешнего хранилища в связи со снижением требований к производительности ввода-вывода. Указанный подход однозначно выигрывает у традиционного подхода,

закключающегося в обеспечении высокой пропускной способности для внешней системы хранения, так как полученная система будет использоваться на полную мощность лишь незначительную часть времени. [124]

Существует также и усовершенствованная версия данного подхода – «Active Burst Buffer». «Active Burst Buffer» проводит пошаговый анализ кэшируемых данных. Данные промежуточного анализа будут храниться в буфере, до того момента, пока сбор данных не будет завершен. После этого промежуточные данные будут агрегированы для получения окончательных результатов анализа. [10]

«Active Burst Buffer» частично похож на MapReduce. MapReduce – это модель распределенной обработки данных, предложенная компанией Google для обработки больших объёмов данных на компьютерных кластерах. Структура MapReduce представлена на рисунке 1.8. [8, 48]

Одно из основных отличий «Active Burst Buffer» от MapReduce состоит в том, что данные в буфере пакетов записываются вычислительными узлами, и трудно убедиться, что файл разделен на блоки фиксированного размера. Таким образом, существующие системы, использующие MapReduce, могут не подходить для анализа на узлах пакетных буферов. [10]

«Active Burst Buffer» обеспечивает следующие функции: 1) быстрый сбор информации от вычислительных узлов, что улучшает производительность ввода-вывода; 2) перенаправление обработанных данных в низкоуровневую параллельную файловую систему для сохранения; 3) создание общего представления о файле для предварительного анализа и обработки; 4) поддержка операции оптимизации (таких как дедупликация и сжатие данных). [10]

Технологию «Active Burst Buffer» схематично можно представить следующим образом (рисунок 1.8):

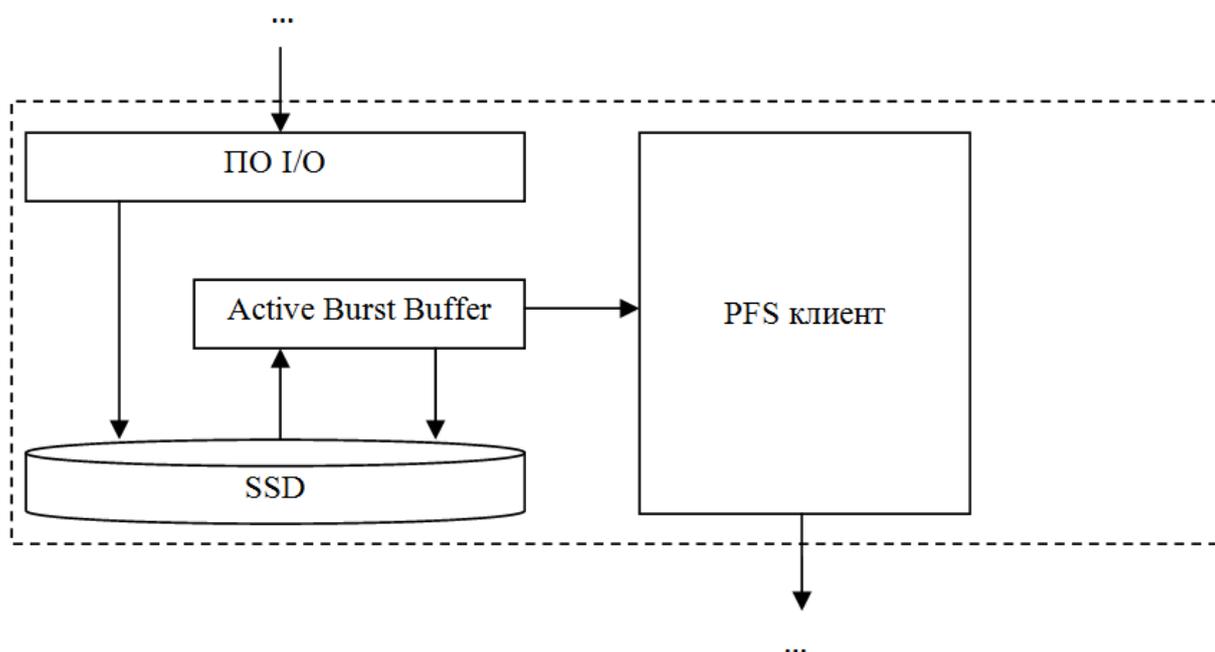


Рисунок 1.8 – Схематичное представление «Active Burst Buffer»

Каждый из таких узлов получает входные данные через промежуточное программное обеспечение ввода-вывода. Активный пакетный буфер («Active Burst Buffer» на рисунке 1.8) работает между этим ПО и клиентом параллельной файловой системы («PFS клиент»). При каждом запросе служба ввода-вывода получает данные и кэширует их на своем локальном SSD через активный пакетный буфер. Затем активный буфер перемещает данные в параллельную файловую систему, а приложения переходят к следующему этапу вычислений. Перед тем, как данные будут записаны в файловую систему, активный буфер может вызвать реализованный пользователем код анализа, если таковой имеется, для анализа кэшированных данных и сохранения промежуточных результатов в определенном файле. Когда сбор всех данных завершен, активный буфер может вызывать другой реализованный пользователем код анализа для агрегирования промежуточных результатов и получения окончательного результата. [10, 38, 67]

В рамках данной диссертационной работы термин большие данные будет преимущественно рассмотрен только в плоскости скорости обработки информации и её объёма. Это обусловлено реализацией проектов на микроконтроллерах IoT Raspberry Pi 3. Данные микроконтроллеры имеют

ограниченные вычислительные мощности и потому тратят достаточно много времени для выполнения тех же задач, что и обычные компьютеры. Проблема объема, прежде всего, вызвана малыми объемами оперативной памяти Raspberry из-за чего даже обычные изображения (более 5 Мб) могут быть достаточно трудными в обработке для микроконтроллера. Запись и чтение файлов из внешней памяти в подобных проектах АСУТП также значительно замедлят процесс обработки информации

### **1.5 Выводы по главе**

В последнее время весьма актуальна проблема машинного зрения. Это обусловлено увеличивающимися темпами автоматизации во всех сферах человеческой деятельности. Подсистемы технического зрения в АСУТП позволяют выполнять задачи без вреда и опасности для здоровья человека. Такие задачи в большом количестве присутствуют на вредных и тяжёлых производствах. Одно из таких – утилизация бытовых отходов. Первый этап данного процесса – сортировка. Большая часть сортировочных конвейеров представляют собой транспортную ленту (по которой движется мусор), устройство анализа объекта и сортирующего механизма. Лидеры рынка применяют в качестве устройства анализа – спектрометры. Эти решения достаточно дорогие.

Альтернативой могут быть системы машинного зрения. Их производительность немного ниже, определяется не сам материал, а ожидаемая физическая форма объекта, подразумевающая использование искомого материала. Но, одновременно с этими недостатками, система получается в разы дешевле и проще. Для снижения влияния недостатков требуется внимательно подойти к выбору алгоритмов распознавания

Существует достаточно большое количество алгоритмов распознавания графической информации. Но наиболее выделяются среди прочих алгоритмы,

основанные на использовании сверточных нейронных сетей. Они демонстрируют выдающееся качество в решении задачи распознавания, но главным препятствием их применения в системах является скорость работы, которой недостаточно для обработки больших объёмов информации.

Для повышения производительности таких систем предложено использовать иерархический подход, основанный на подходе «Coarse-to-Fine». Поиск будет производиться последовательно и итерационно, постепенно уточняя область поиска. Это позволит снизить нагрузку на систему в каждый из моментов времени активного поиска, что положительно скажется на общей производительности.

Для повышения эффективности использования вычислительных ресурсов были рассмотрены способы использования иерархической архитектуры для минимизации задержек при передаче информации. А обеспечить надлежащую пропускную способность позволяет иерархический подход, уменьшающий количество вычислительных операций микроконтроллером.

## **Глава 2. Иерархический метод обработки растровых изображений в автоматизированных системах управления с применением модулей машинного зрения**

### **2.1 Сравнение иерархического и других подходов в АСУТП**

Решать проблемы нехватки вычислительных ресурсов и недостаточной пропускной способности обычно стараются изменением архитектуры системы – централизованные и децентрализованные системы. Данные подходы называются Cloud computing и Edge computing.

Cloud computing (далее облачные вычисления) – модель обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых вычислительных ресурсов, которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами или обращениями к провайдеру. [46]

Другими словами, облачные вычисления являются примером централизованных систем обработки данных. При их использовании потребители информационных технологий могут существенно снизить капитальные расходы – на построение центров обработки данных, закупку серверного и сетевого оборудования, аппаратных и программных решений по обеспечению непрерывности и работоспособности – так как эти расходы поглощаются провайдером облачных услуг. Провайдер предоставляет необходимое количество вычислительной мощности для обработки запроса пользователя. Однако стоит отметить, что пользователю придётся передать обрабатываемые данные третьему лицу, что не всегда является безопасным.

Схожим по идее функционирования является направление Green computing (далее зелёные вычисления) с отличием в том, что зелёные вычисления – это исследование и практика экологически устойчивых вычислений. Основная идея

построена на требовании снижения энергопотребления вычислительным оборудованием (чем меньше тратится энергии, тем меньше энергопоставщик использует ресурсов и наносит вреда природе). На практике это направление представляет собой объединение серверов. То есть отказ от большого количества серверов в своей организации, и перенос максимального числа задач на один общий сервер. Энергия сберегается на питание только одного сервера, а также нужно меньше энергии на кондиционеры для их охлаждения. В отличие от облачных вычислений, зелёные вычисления лишены недостатка передачи данных третьим лицам, но в тоже время из-за этого часто бывают менее экономически выгодными. [93, 126,]

Централизация системы направлена решение проблемы нехватки вычислительной мощности и увеличение экономической эффективности. Однако оно не решает проблемы пропускной способности, а иногда даже усугубляет её.

Другой вариант решения проблем – децентрализация системы – Edge computing. Самое важное отличие Edge computing от облачных вычислений заключается в том, что сбор и анализ данных проводится не в централизованной вычислительной среде, такой как ЦОД, а в том месте, где происходит генерация потоков данных. Источниками данных служат цифровые устройства (не обязательно находящиеся в одной локации), которые затем передают эти данные в режиме реального времени (зависит от ситуации, передача информации может быть отложена) в центральный репозиторий. [119]

Примечательно, что некоторые IT-специалисты не различают разницу между «граничными» и традиционными распределенными вычислениями, которые предназначаются для распараллеливания вычислительных мощностей между ЦОДами и офисами, и существуют несколько десятилетий. Разница между ними состоит в том, что Edge computing привязан к данным IoT, которые собираются с удаленных датчиков, смартфонов, планшетов и машин.

Такой подход, несомненно, положительно сказывается на пропускной способности основной системы, но является экономически затратным (для поддержания эффективности и скорости обработки на том же уровне, что и при

традиционном подходе). Это обусловлено небольшими вычислительными возможностями устройств IoT.

Мы предлагаем обратить внимание на подход направленный на изменение принципа анализа изображения – иерархический подход. Он может сочетаться как с централизованным, так и с децентрализованным подходом. Он основан на поэтапном уточнении местоположения распознаваемого объекта на изображении.

В случае «централизованного» иерархического подхода, вначале анализируется «большое» изображение в низком разрешении – результатом является набор областей изображения, на которых может находиться интересующий нас объект. Следующим шагом анализируются обнаруженные области в уже большем разрешении. В итоге система получает либо подтверждение наличия объекта в области, либо уточнённую область, где может находиться объект. Иерархия таких шагов позволяет значительно снизить количество используемых вычислительных ресурсов и соответственно система на более дешёвых и менее энергозатратных компонентах.

В случае «децентрализованного» иерархического подхода, рационально использовать «умную камеру». Данная камера может находиться в двух режимах – режиме ожидания и режиме записи. Во время режима ожидания камера снимает происходящее в низком разрешении, но достаточном для распознавания появления в кадре объекта. Это первичное распознавание будет выполняться посредством дешёвого микроконтроллера и простого программного кода, реализованного на Python. Как только камера обнаружит появление объекта в кадре, она переключается в режим записи, отправляя информацию серверу для установления личности человека, где уже используется нейросетевое распознавание. Таким образом, данное устройство сильно снижает нагрузку на сервер за счёт уменьшения количества поступающей информации. Для ещё более ощутимого эффекта программный код, отвечающий за обнаружение объекта в кадре можно модифицировать таким образом, чтобы на сервер поступало не всё изображение, а только лишь ROI.

## 2.2 Алгоритм иерархической обработки растровых изображений

Основной идеей метода является последовательное повышение разрешения качества изображения, ограничивая распознаваемое изображение границами области интереса (ROI).

Метод заключается в последовательном выполнении следующих действий:

1) Создаются дубликаты исходного изображения в более низком качестве (меньшее разрешение, меньшее число пикселей); [127]

2) Изображения дробятся на фрагменты и сохраняются в специальной файловой системе, помечаемые необходимой поисковой информацией; [127]

3) Производится распознавание дубликата изображения (полученного после 1 действия) – грубый поиск. Алгоритм, применяемый на этом шаге, должен быть равен или иметь более низкий показатель вычислительной сложности по сравнению с основным алгоритмом применяемом на следующем шаге, но позволяющий уверенно отличать объект распознавания от фона. На изображении отмечаются ROI которые удовлетворяют критериям грубого поиска. По координатам ROI определяются фрагменты изображения большего разрешения, которые необходимо проанализировать. Данный этап повторяется до тех пор, пока следующими фрагментами не являются фрагменты исходного изображения; [127]

4) Происходит распознавание фрагментов исходного изображения – точный поиск. Для выполнения точного поиска применяется алгоритм с наилучшим качеством распознавания (удовлетворяющий по количеству ошибок первого и второго рода). [127]

Для всех этапов метода рекомендуется применять фон контрастный к объекту.

Как показал анализ, алгоритмы сжатия и дробления изображения на фрагменты не оказывают сильного влияния на распознавание описанном в 3 шаге, т.к. алгоритм поиска, в данном случае, используется для обнаружения

местоположения объекта на изображении. Количество выполнений 3 шага напрямую связано с доступными вычислительными ресурсами системы, и может быть представлено как одна итерации, так и несколько.

Вычислительная сложность алгоритма зависит от применяемых в нём методов обработки информации. Для алгоритмов использующих операцию свёртки в общем случае вычислительную сложность можно определить как (1):

$$F_{\text{расп}} = V * (\alpha + \beta + \gamma), \quad (1)$$

где  $V$  – размер изображения,  $\alpha$  – количество тактов процессора, требуемое для выполнения сложения и вычитания при расчёте одного «скользящего окна»,  $\beta$  – количество тактов процессора, требуемое для выполнения умножения и деления при расчёте одного «скользящего окна»,  $\gamma$  – количество тактов процессора, требуемое для выполнения вычисления экспоненты при расчёте одного «скользящего окна».

Другими словами вычислительная сложность алгоритма можно оценить как линейную функцию, т.к. алгоритму требуется перебрать все  $V$  пикселей изображения.

Свёрточные алгоритмы чаще всего применяются в нейронных сетях, где сложность уже нелинейная и зависит от количества слоёв и нейронов в них. Однако суть задачи остаётся прежней – сокращение числа информации на входе в систему. Иерархический метод даст положительные результаты тогда и только тогда, когда будет выполняться условие (2):

$$(V_{\text{грубый поиск}} + V_{\text{точный поиск}}) < V_{\text{обычное распознавание}} \quad (2)$$

где  $V_{\text{грубый поиск}}$  – размер дубликата изображения, распознаваемого на 3 шаге метода,  $V_{\text{точный поиск}}$  – суммарный размер ROI, выбранных для распознавания на 4 шаге метода,  $V_{\text{обычное распознавание}}$  – размер исходного изображения.

На рисунке 2.1 представлен график расчётного времени, необходимого для обработки одного кадра (определённого разрешения) нейронными сетями MobileNet и LeNet. При неизменной вычислительной сложности (время

распознавания полного изображения получено экспериментально) с уменьшением объёма распознаваемой информации (ROI) наблюдается общая тенденция – время необходимое для распознавания одного кадра сокращается. При увеличении пропускной способности системы данная тенденция сохраняется.

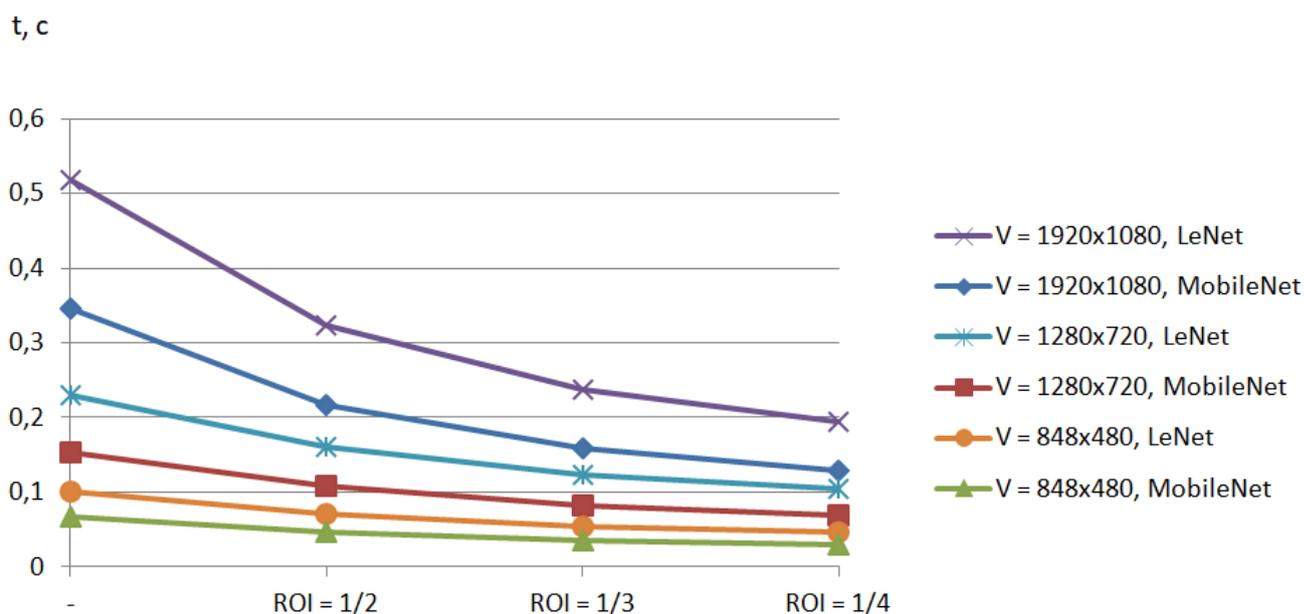


Рис. 2.1. График расчётного времени для распознавания одного кадра

Эти расчёты применимы, когда сервер не оказывается перегруженным. Основываясь на законах теории производительности вычислительных систем, время обработки изображения можно приближённо рассчитывать как линейную функцию только пока коэффициент использования (загруженности системы) меньше 70%. При перегрузе сервера и образовании очереди из изображений, время распознавания изменяется нелинейно и стремится к бесконечности.

Таким образом, иерархический метод позволяет решить обе основные проблемы систем распознавания объектов на изображении, сокращая объём обрабатываемой информации – уменьшается количество требуемых вычислительных ресурсов и общее время, необходимое алгоритму для обработки изображения.

Данный метод имеет преимущество над стандартными по времени распознавания исходного изображения. Как правило, для анализа изображения (стандартными подходами) требуется обойти «скользящим окном» все пиксели изображения. Сложность такой операции зависит от размеров исходного

изображения – время распознавания (в грубом представлении) является произведением трёх чисел: высоты изображения в пикселях, ширины изображения в пикселях и времени требуемом на операции распознавания для каждой итерации «скользящего окна». Рассмотренный выше подход позволяет значительно уменьшить первые два числа (высота изображения в пикселях, ширина изображения в пикселях), что снижает общее время обработки. Однако следует помнить, что каждая итерация этапа грубого поиска требует некоторого времени. Для получения эффекта ускорения процесса распознавания необходимо использовать достаточное количество итерация грубого поиска. В противном случае суммарное время распознавание изображение может быть сопоставимым с временем показываемым стандартными подходами.

Следующий график (рис. 2.2) демонстрирует оценочные прямые зависимости времени обработки, от разрешения исходного изображения. Линия стандартного подхода построена для ситуации, когда изображение анализируется скользящим окном, затрачивающим фиксированные 2 условные единицы времени на один шаг.

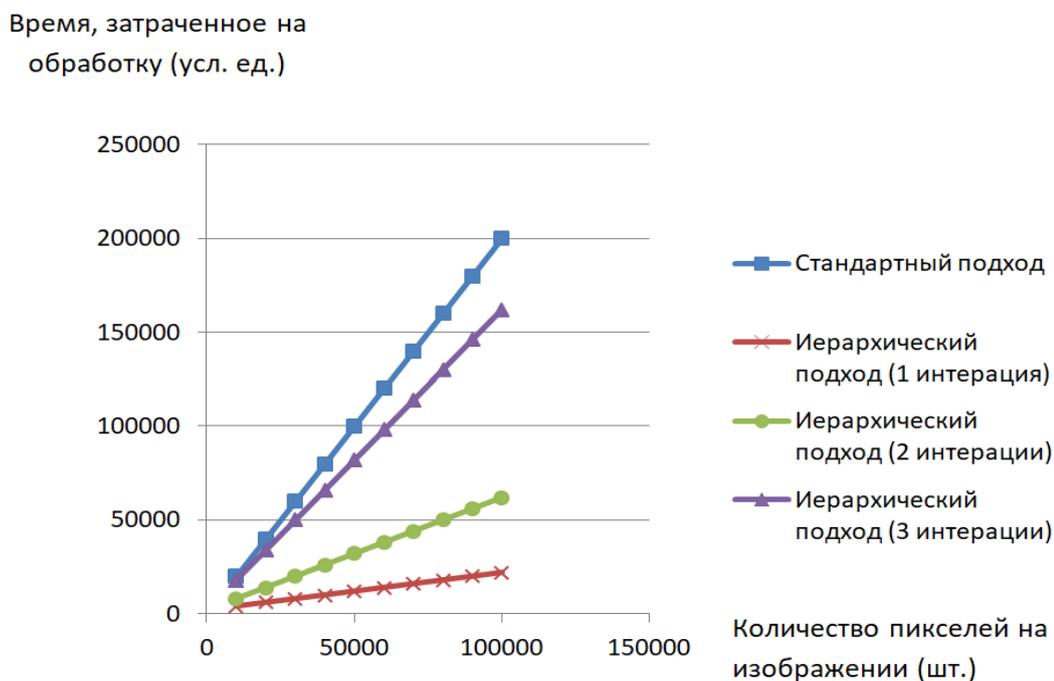


Рисунок 2.2 – График зависимости времени обработки изображения от количества пикселей на изображении

Линия иерархического подхода (1 итерации) построена для ситуации, когда изображение анализируется скользящим окном, затрачивающим 2 условные единицы времени на один шаг, но при грубом поиске анализируется изображение, сжатое в 10 раз, а при точном поиске оценивается фрагмент размером 1000 пикселей. Линия иерархического подхода (2 итерации) построена аналогично предшествующей, но на этапе грубого поиска сначала анализируется изображение, сжатое в 10 раз, а потом для уточнения результата оценивается фрагмент изображения сжатый в 5 раз. Линия иерархического подхода (3 итерации) построена аналогично предшествующей, но на этап грубого поиска состоит из анализа 3 изображений: сжатого в 10, 5 и 2 раза соответственно.

Подразумевается, что все вычисления, представленные выше, проводятся на одинаковом оборудовании с использованием одинаковых алгоритмов распознавания. Соответственно, при распознавании используются одинаковые низкоуровневые операции, что позволяет использовать универсальную формулу расчёта времени. Эта формула может быть применена в подобных случаях для любой вычислительной платформы, занимающейся визуальным распознаванием, и для любого по сложности алгоритма распознавания объекта на изображении.

Также, данный метод может быть применён в областях, где имеются ограничения по объёму информации, передаваемой серверу (окончательная обработка происходит на сервере – централизованный иерархический подход). Благодаря первичному обнаружению контейнеров на ленте имеется возможность выделения ROI и отправка только его в качестве исходного изображения. Другими словами во время, когда конвейер пуст, камера не будет передавать серверу изображение для распознавания. Когда на ленте появляется контейнер, камера его обнаруживает, рассчитывает его границы и начинает отправлять серверу только ту часть изображения, где присутствует контейнер. Практика показала, что при постоянной нагрузке на конвейер (контейнеры идут один за другим без перерывов) экономия трафика составляет в среднем от 30% и более. А если нагрузка на конвейер носит импульсный характер (между появлением контейнеров на ленте имеется время, когда конвейер пуст) то трафик может

сократиться в разы. В тоже время не стоит забывать, что снижение количества трафика (информации, которой придётся обработать серверу) положительно сказывается на нагрузке не только сети, но и сервера. Этот факт позволяет при тех же условиях использовать большее количество камер передающих информацию, либо повысить общую скорость обработки информации сервером за счёт наличия освободившихся вычислительных ресурсов.

## **2.3 Применение иерархического подхода к существующим проектам**

### **2.3.1 Распознавание статических объектов на изображениях**

Reverse vending machine (автомат по приёму тары, RVM) – это автомат, позволяющий людям возвращать пустые контейнеры (бутылки ПЭТ и алюминиевые банки) для напитков за вознаграждение. [1]

Большинство машин используют контроль материала контейнера (например, с помощью ИК-спектрометра, как и на конвейерах), контроль формы контейнера, контроль штрих-кода для проверки тары. Эти три основные процедуры контроля делают любую попытку мошенничества совершенно невозможной, но в то же время делает RVM слишком дорогостоящим. [16, 35] Другой метод идентификации, основанный на распознавании объектов с нейронными сетями, используя контроллер IoT (Raspberry PI 3) с экраном, камерами и некоторыми датчиками [31] значительно снижает себестоимость автомата, но создаёт ряд трудностей связанных с процессами обработки изображения, которые были уже представлены в разделе 2.1.

Основным инструментом распознавания в представленном RVM является сверточная нейронная сеть (Convolutional neural networks, CNN). Данный вид сетей сейчас считают самым мощным классификатором изображения в области машинного зрения. [3, 26, 50] Реализация активации нейронной сети выполнена

как скрипт Python, который загружается после срабатывания датчика появления контейнера в отсеке идентификации и получения снимка от установленной там камеры. Сбор информации о распознаваемой таре осуществляет набор камер. В предложенном варианте конструкции реализован оптимальный вариант с установкой двух камер. Данное количество может быть изменено в случаях технической необходимости так, чтобы общая область видимости, у всех включенных в набор камер, была не меньше площади отсека распознавания. Вариант с одной камерой не является оптимальным для такой конструкции, так как из-за увеличения высоты отсека распознавания (камеру необходимо закрепить выше над «полом» отсека так, чтобы контейнер обязательно целиком находился в кадре) увеличивается и размер самого RVM, что является критичным для многих потенциальных заказчиков.

Устройство работает следующим образом. Пользователь помещает предмет через приёмное отверстие в отсек распознавания. Видеокамеры фиксируют появление предмета в отсеке распознавания и передают изображение микроконтроллеру. Микроконтроллер, зафиксировав нахождение предмета в отсеке распознавания целиком, собирает изображения, полученные со всех установленных в автомат камер, склеивает их, получая полное изображение предмета. Далее микроконтроллер с помощью нейронной сети распознаёт предмет. В случае принятия решения, что в отсеке распознавания находится ПЭТ (полиэтилентерефталат) бутылка, посылается сигнал на одно из запирающих устройств, которое открывает одну из створок и бутылка скатывается в отсек хранения бутылок. В случае принятия решения, что в отсеке распознавания находится алюминиевая банка, посылается сигнал на второе запирающее устройство, которое открывает другую створку – банка скатывается в отсек хранения банок. В случае принятия решения, что в отсеке распознавания находится непринимаемый контейнер (или иной предмет, не являющийся контейнером), посылается сообщение на дисплей с просьбой к пользователю изъять предмет из отсека распознавания. После завершения всех операций все

элементы переходят к стартовому состоянию и ожидают нового появления предмета в отсеке распознавания.

Применяемая нейронная сеть обучается на фотографиях банок и ПЭТ бутылок различных форм и определяет основные функции, необходимые для классификации. Благодаря этой особенности любое новое изображение ПЭТ бутылки может быть идентифицировано как совокупный класс ПЭТ бутылок, если контейнер обладает соответствующими свойствами. Банки обрабатываются аналогичным образом. Даже если контейнер механически поврежден и имеет нестандартную форму, то он может быть идентифицирован как один из описанных классов. [31] Согласно тестам, распознавание тары при помощи нейронных сетей является более быстрым по сравнению с заявленными характеристиками RVM. В ходе тестирования автомата была настроена нейронная сеть, способная идентифицировать тару по 3 классам – «бутылка ПЭТФ», «алюминиевая банка», «другое». [97, 98]

Однако производители часто заключают контракты на переработку своей тары, что не позволяет полноценно использовать предложенное решение в автоматах (в RVM технически данная функция возможна, но тоже не заявлена). Поэтому было принято решение о модификации предложенного отсека распознавания и расширении функционала автомата. Учитывая особенности текущего законодательства в области утилизации отходов, возможны два пути решения – классификация тары с использованием этикетки или с распознаванием штрихкодов конкретного артикула продукта [35, 109], которые бы дополняли распознавание формы с помощью нейронных сетей.

Реализация первого варианта возможна с использованием ранее разработанной нейронной сети, но не вписывается в концепцию разрабатываемой системы (любое новое изображение ПЭТ бутылки или алюминиевой банки может быть идентифицировано как совокупный класс, если контейнер обладает соответствующими свойствами и механически поврежденный контейнер, имеющий нестандартную форму или без некоторых частей (в том числе с утратой этикетки) может быть идентифицирован как один из описанных классов). Кроме

того, производители часто меняют дизайн контейнеров (цвет и изображение на этикетке), что потребует частое техническое обслуживание автомата для обновления образов распознаваемых классов, что негативно скажется на итоговой себестоимости автомата и приведет к нерентабельности данного проекта.

Для второго варианта необходимо наличие неповрежденной этикетки (или фрагмента этикетки со штрихкодом), но данный вариант не противоречит предлагаемой идее и конструкции автомата, и прост в реализации. Кроме того, штрихкод постоянен в течение продолжительного времени для производителя на всем его жизненном цикле.

Штрихкод – графическая информация о товаре в виде последовательности штрихов и пробелов различной ширины и нескольких цифр, расположенных под ними. Код наносится на поверхность самого изделия, на этикетку или упаковку. Задача считывания этого штрихкода может заключаться либо в распознавании цифр под полосками, либо самих полосок. Существует достаточно большой ряд алгоритмов, позволяющих проводить данные манипуляции. Это могут быть как специализированные библиотеки (Zbar), так и нейронные сети.

ZBar – это библиотека с открытым исходным кодом для считывания штрихкодов из видеопотока или отдельных изображений. На рисунке 2.3 приведен пример распознавания штрихкода. Программа может отобразить рамки распознанного штрихового кода и вывести считанную информацию. Полученная числовая последовательность проверяется на соответствие акционной базе. В случае обнаружения совпадения, автоматом принимается решение о выдаче вознаграждения.

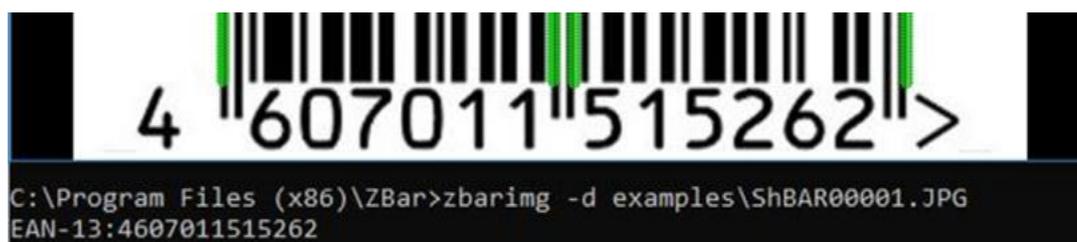


Рисунок 2.3 – Пример распознавания 13 разрядного европейского штрихкода EAN-13.

Иной подход можно реализовать в нейронных сетях. На данный момент существует уже много решений позволяющих считывать печатные символы (в том числе и на базе TensorFlow, применяемой в нашем автомате [31, 97, 98]). Преимуществом такого решения будет возможность считывать поврежденные штрихкоды, восстанавливая нераспознанные символы с некоторой степенью вероятности. Восстановленная таким образом информация может считаться верной, если результат распознавания нейронной сетью имеет отметку о высокой вероятности, и при совпадении суммы знаков распознанного штрихкода с контрольным числом в конце кода.

Однако, подобные решения очень требовательны к качеству считываемого изображения – не должно быть лишних деталей, штрихкод должен быть расположен в фокусе, быть достаточно большим [121]. Данную проблему можно решить двумя путями – либо предложить пользователю автомата помещать акционную тару в тароприёмник, соблюдая строгую ориентацию штрихкода по отношению к считывающему устройству, либо сделать автоматическую подсистему нахождения штрихкода на таре. Первый подход достаточно прост, но может вызвать ряд неудобств у пользователя, чем вызовет негативное отношение к проводимой акции. Второй подход осложнен тем, что размещение штрихкода на таре не регламентировано и потому производитель может его разместить в любом месте и под любым углом к оси вращения тары. Кроме того, изображение кода тоже может быть видоизменено (примеры представлены на рисунке 2.4). Однако штрихкоды всегда помещаются на белом прямоугольном фоне, и этот признак можно использовать для усовершенствования работы предложенного алгоритма.



Рисунок 2.4 – Примеры нестандартных вариантов штрихкода EAN-13.

Для этого этапа многоуровневого распознавания логично использовать быстрый и нетребовательный алгоритм. В его работе допустимы ошибки, так как он выполняет только грубый поиск объекта на изображении. Поэтому был выбран алгоритм поиска расположения штрих-кода на основе каскадов Хаара, который выполняет поиск прямоугольного белого объекта с полосками [47, 74, 94]. Полученные в результате работы каскада Хаара области интереса (ROI) можно передавать для детального изучения следующему уровню распознавания.

Суть модификации автомата по приему тары заключается в поэтапном иерархическом уточнении информации о распознавании тары. Конструкция приемного отделения автомата будет оснащена валиком для поворота принятой тары (в горизонтальном исполнении отсека) или поворотной площадкой (в вертикальном исполнении отсека).

Алгоритм распознавания преобразуется к следующему виду:

1) Процесс обнаружения тары в отсеке распознавания и фиксация ее формы является начальным шагом в данном алгоритме. До того момента, пока автомат не обнаружит тары в отсеке распознавания он находится в режиме ожидания [97, 98].

2) Проверка принадлежности тары к одному из классов («бутылка ПЭТФ» или «алюминиевая банка»). Данная проверка проводится нейронной сетью [31, 97, 98]. Если распознаваемый объект имеет слишком малый процент принадлежности к описанным классам, то он возвращается пользователю, а автомат на дисплее отображает соответствующую информацию и возвращается к шагу 1.

3) Если в классе, определенном на шаге 2, нет тары, принимаемой на особых условиях, то автомат перемещает распознанную тару в отсек хранения и выдает пользователю соответствующую информацию, переходя к шагу 1. В обратном случае автомат переходит к шагу 4, запуская дополнительный этап иерархического распознавания штрих-кода, описываемый в данной статье.

4) Происходит плавное вращение тары в отсеке распознавания для создания комплексного изображения боковой поверхности тары, путем

программной склейки изображений узких участков поверхности, находящихся перед видеокамерой (для минимизации искривлений на итоговом изображении).

5) С помощью каскадов Хаара на изображении, полученном на шаге 4, выделяются области соответствующие искомым признакам (прямоугольный белый объекта с полосами контрастного цвета внутри). Пример такого грубого поиска продемонстрирован на рисунке 2.5.



Рисунок 2.5 – Пример грубого поиска штрихкода на упаковке тары.

б) Отмеченные области (их может быть несколько на одной таре из-за особенностей графического дизайна тары), полученные на шаге 5, передаются нейронной сети для распознавания и в случае успешного считывания правильного штрихового кода автомат принимает решение о принятии акционной тары. Пользователю сообщается о принятии таковой тары и выдается вознаграждение. В случае, если распознанный штрихкод не соответствует ни одному, хранимому в базе акционной тары, то тара принимается также, как описано в шаге 3. Далее автомат возвращается к шагу 1.

Описать данный алгоритм можно с помощью модели (рисунок 2.6), где:

- S1 – состояние ожидания системы (пункт 1 из предшествующего алгоритма);
- S2 – состояние активации системы (пункты 2 и 3 из предшествующего алгоритма);

- S3 – состояние уточнения информации (пункт 4-6 из предшествующего алгоритма);

Другими словами модернизация добавляет S3 к стандартному графу состояний. При попытке анализа штрихкода в стандартном (немодернизированном) варианте автомата (при условии попадания штрихкода в объектив) потребовались бы все вычислительные ресурсы микрокомпьютера и достаточно длительное время. В модернизированном варианте во втором состоянии будет анализироваться сильно сжатое изображение, а в третьем – небольшой фрагмент большого изображения. Даже при относительно равном времени выполнения алгоритмов модернизированный метод будет не хуже стандартного.

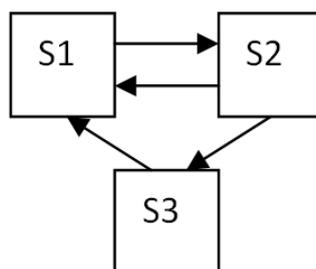


Рисунок 2.6 – Схема автомата состояний модернизированного автомата по приёму тары

Таким образом, данный подход актуален для условий ограниченных вычислительных мощностей микроконтроллера, применяемого в автомате [10]. Он позволяет избежать большого количества операций по распознаванию штрихкода, в случае если бы применялся режим поиска штрихкода в реальном времени, заменяя их единичными вызовами простых алгоритмов. Происходит экономия ресурсов памяти, т.к. наиболее ресурсоемкая часть (распознавание штрихкода) выполняется не над большим комплексным изображением боковой поверхности тары, а над его отдельными фрагментами. Последний факт также положительно сказывается на эффективности и времени распознавания.

Недостатками данного подхода являются усложнение конструкции отсека распознавания механизмом вращения тары и необходимость затраты дополнительного времени на вращение тары. Этот недостаток можно устранить, применив систему зеркал, дающих камере обзор на тару со всех сторон в боковой проекции. Однако в таком случае возникнет проблема очистки зеркал от возможного загрязнения. Поэтому описанный подход считается применимым для автомата по распознаванию тары.

### **2.3.2 Распознавание динамических объектов на изображениях**

В RVM контейнеры помещаются в отсек, и только после этого происходит распознавание. В условиях конвейера распознавание должно происходить постоянно, так как неизвестно в какой момент времени на ленте окажется контейнер. Это вносит иные трудности в разработку автомата по сортировке. Одним из решений может быть SmartCamera.

SmartCamera – это камера, выполненная на базе микроконтроллера IoT (Raspberry PI 3), способная проводить предварительную обработку передаваемого серверу изображения и выполнять некоторые логические функции. Архитектура камеры сопоставима по стоимости со сложными камерами, оснащёнными рядом дополнительных датчиков, но имеет большой потенциал по настройке системы. Такая настройка может заключаться в наложении «умных» фильтров (автоматически подстраивающихся под изменяющиеся условия съёмки), динамическом изменении качества и масштаба изображения передаваемого серверу.

Так как установка камеры предполагается в постоянных условиях (камера жёстко закреплена над лентой, освещение и другие внешние условия можно считать постоянными) появляется возможность применить выделение статичного фона. Для этого камера должна выполнить калибровку при запуске сделав ряд снимков и придя к выводу, что перед ней нет движущихся объектов. Это достигается последовательным вычитанием нового изображения из

предшествующего. Как только числовые значения всех пикселей двух изображений станут равными или их отличия не будут превышать определённого порога, камера запоминает последнее изображение. Оно будет считаться «статичным фоном» или сценой распознавания (рисунок 2.7 а). После этого шага камера переходит в режим ожидания объекта, продолжая съёмку. Все последующие изображения будут вычитаться из статичного фона. И как только дешёвый микроконтроллер обнаруживает пиксели отличные от чёрного (рисунок 2.8 а), он может переходить в режим подготовки изображений для отправки.



Рисунок 2.7 – Пример изображений с камер, установленных над конвейером (а – пустой конвейер, б – ПЭТ бутылка на конвейере)

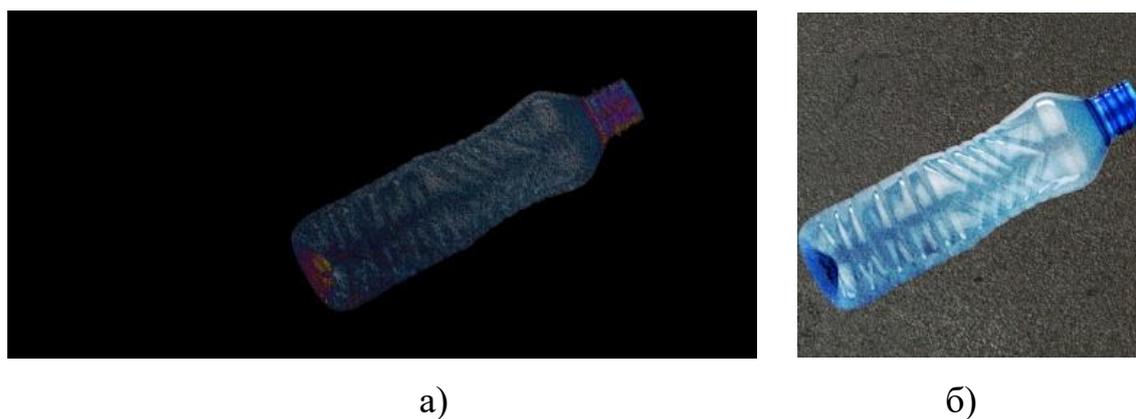


Рисунок 2.8 – Пример изображений, обработанных (а – вычитание объекта из сцены распознавания, б – ROI определённое камерой)

При вычитании микроконтроллер получает маску ROI. В маске определяются крайние координаты объекта, которые в дальнейшем используются

для вырезания ROI (рисунок 2.8 б) из исходной фотографии (рисунок 2.7 б). В данном случае, так как ориентация предмета на ленте может быть произвольной, изображение вырезается поперёк движения конвейера, но во всю его ширину. В случае если микроконтроллер получает маску закрывающую поле зрения (остаётся мало чёрных пикселей), то камера переходит в режим калибровки и делает ряд снимков, дожидаясь появления нового «статичного фона». Это может происходить при смене освещения, появления в кадре крупногабаритных объектов, запотевании объективов.

Реализация подобного фильтра возможна без применения сложных библиотек (OpenCV), а лишь с помощью стандартных библиотек Python, отвечающих за математические действия (matplotlib, numpy) и работу с изображениями (skimage). Это позволяет применить данное решение в условиях реализации умной камеры на базе дешёвого микроконтроллера. Такой «умный фильтр» позволяет значительно снизить трафик, передаваемый на сервер – исходное изображение имело размер 1 МБ (рисунок 2.7 б), то после проделанной операции размер уменьшился в среднем до 0,6 МБ (рисунок 2.8 б).

## 2.4 Выводы по главе

Решать проблемы нехватки вычислительных ресурсов и недостаточной пропускной способности обычно стараются изменением класса системы – централизованные и децентрализованные системы. Предложено отступить от данного подхода и обратить внимание на изменение принципа анализа изображения – иерархический подход. Он может сочетаться как с централизованным, так и с децентрализованным подходом. Иерархический подход основан на поэтапном уточнении местоположения распознаваемого объекта на изображении.

Создаются дубликаты исходного изображения в более низком качестве (меньшее разрешение). На первом этапе распознавания производится на

оригинальном изображении со значительно сниженным разрешением – отмечаются ROI, которые удовлетворяют критериям грубого поиска. На втором этапе происходит обработка фрагментов исходного изображения в стандартном режиме распознавания – точный поиск.

Данный подход имеет преимущество над стандартными по времени распознавания исходного изображения и по ресурсозатратности, так как позволяет значительно уменьшить разрешение обрабатываемого изображения, что снижает и общее время обработки. Однако следует помнить, что каждая итерация этапа грубого поиска требует некоторого времени. Для получения эффекта ускорения процесса распознавания необходимо использовать достаточное количество итерация грубого поиска. В противном случае суммарное время распознавание изображение может быть сопоставимым с временем показываемым стандартными подходами и даже больше.

## Глава 3. Моделирование процесса иерархического хранения и обработки растровых изображений

### 3.1 Описание условий моделирования процесса

Для корректной оценки скорости работы такой системы необходимо использовать закон Литтла (англ. Little's law) (3) к вычислительным ресурсам платформы. [2, 54] В теории массового обслуживания, разделе теории вероятностей, законом Литтла называют сформулированную американским учёным Джоном Литтлом теорему: Долгосрочное среднее количество заявок  $L$  в стационарной системе равно долгосрочной средней интенсивности  $\lambda$  входного потока, умноженной на среднее время  $W$  пребывания заявки в системе:

$$L = \lambda \cdot W \quad (3)$$

Среднее время обработки каждой отдельной заявки в такой системе будет зависеть от загруженности сервера. Из фундаментальных законов производительности вычислительных систем, известна зависимость среднего времени обработки заявки от состояния загруженности системы в конкретный момент времени (коэффициент использования (4)). [2] Когда коэффициент использования достигает 100%, то есть сервер оказывается перегруженным, время обработки стремится к бесконечности из-за ограниченности вычислительных ресурсов. На рисунке 3.1 представлен график зависимости времени обработки запроса от загруженности сервера.

Компонент, испытывающий перегрузки, вынуждает систему выстраивать заявки на обработку информации в очередь, означающую, что каждый последующий запрос должен ожидать полного завершения предшествующего ему для получения необходимого количества вычислительных ресурсов. Согласно графику, представленному на рисунке 2, точкой перегиба можно считать отметку, где коэффициент использования принимает значение 70%. При меньших

показателях увеличение времени обработки можно считать приемлемым. При больших – увеличение времени обработки происходит слишком быстро.

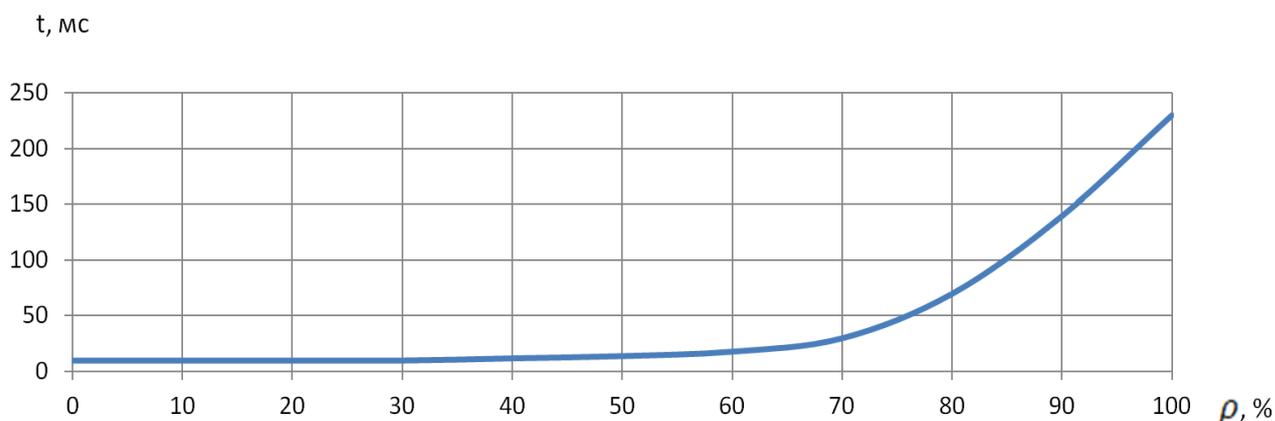


Рисунок 3.1 – Влияние загрузки сервера на время обработки заявки

Таким образом, для обеспечения стабильной продуктивной работы системы необходимо поддерживать коэффициент использования менее 0,7. В остальных случаях модель может быть не корректной, так как время обработки будет описываться нелинейной функцией, зависящей от множества параметров (программное и аппаратное обеспечение системы, техническое состояние аппаратных компонентов, динамические показатели исправности системы и прочее) и уникальной для каждого конкретного случая.

Коэффициент использования рассчитывается следующим образом:

$$\rho = \frac{\lambda}{\mu} \quad (4)$$

где  $\lambda$  – интенсивность потока заявок, а  $\mu$  – интенсивность обработки заявок системой.

Условная система для моделирования будет состоять из камеры (источника заявок) снимающих полотно транспортной ленты, по которой движутся объекты, нуждающиеся в распознавании. Подразумевается, что вычислительная платформа выбрана таким образом, чтобы коэффициент использования был ниже 0,7 в момент 1 полного изображения, получаемого с камеры (источника заявок). Т.е. система всегда работает без перегрузки и колебаниями времени необходимого на

выполнение операций можно пренебречь. Считается, что вычислительная платформа не выполняет более никаких функций и расходует вычислительные ресурсы только на процесс распознавания. Время передачи информации от камер (источников заявок) считается бесконечно малым.

## 3.2 Аналитическая модель процесса иерархической обработки растровых изображений

### 3.2.1 Описание процесса создания аналитической модели

Данную систему можно представить как  $n$ -канальную СМО с отказами. Отказы в данном случае будут представлять собой объекты, на которые система не успела среагировать из-за того, что вычислительные возможности были заняты распознавание другого объекта. [85, 86] Рассмотрим систему подобного типа, оставаясь в рамках марковской схемы. Предположим, система  $S$  (СМО) одноканальная и имеет два состояния:  $S_0$  — канал свободен,  $S_1$  — канал занят. Размеченный граф состояний представлен на рис. 3.2.

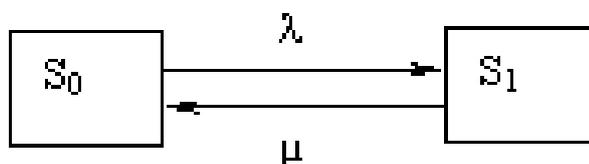


Рисунок 3.2 – граф состояний одноканальной СМО с отказами

В предельном, стационарном режиме система алгебраических уравнений для вероятностей состояний имеет вид (4):

$$\begin{cases} \lambda p_0 = \mu p_1, \\ \mu p_1 = \lambda p_0, \end{cases} \quad (4)$$

т.е. система вырождается в одно уравнение. Учитывая нормировочное условие  $p_0+p_1=1$ , найдем из (3) предельные вероятности состояний (5):

$$p_0 = \frac{\mu}{\lambda + \mu}, p_1 = \frac{\lambda}{\lambda + \mu}, \quad (5)$$

которые выражают среднее относительное время пребывания системы в состоянии  $S_0$  (когда канал свободен) и  $S_1$  (когда канал занят), т.е. определяют соответственно относительную пропускную способность  $Q$  системы (6) и вероятность отказа  $P_{\text{отк}}$  (7):

$$Q = \frac{\mu}{\lambda + \mu}, \quad (6)$$

$$P_{\text{отк.}} = \frac{\lambda}{\lambda + \mu}. \quad (7)$$

Абсолютную пропускную способность найдем, умножив относительную пропускную способность  $Q$  на интенсивность потока отказов (8) [3]

$$A = \frac{\lambda\mu}{\lambda + \mu}. \quad (8)$$

Если система многоканальная, то  $S$  имеет следующие состояния:  $S_0, S_1, S_2, \dots, S_k, \dots, S_n$ , где  $S_k$  — состояние системы, когда в ней находится  $k$  заявок (т.е. занято  $k$  каналов). [85, 86] Граф состояний СМО представлен на рис. 3.3.

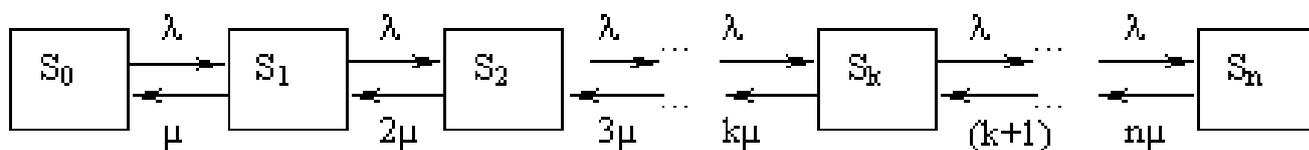


Рисунок 3.3 – граф состояний  $n$ -канальной СМО с отказами

Поток заявок последовательно переводит систему из любого левого состояния в соседнее правое с одной и той же интенсивностью  $\lambda$ . Интенсивность же потока обслуживаний, переводящих систему из любого правого состояния в соседнее левое состояние, постоянно меняется в зависимости от состояния. Действительно, если СМО находится в состоянии  $S_2$  (два канала заняты), то она

может перейти в состояние. S1 (один канал занят), когда закончит обслуживание либо первый, либо второй канал, т.е. суммарная интенсивность их потоков обслуживании будет  $2\mu$ . Аналогично суммарный поток обслуживаний, переводящий СМО из состояния S3 (три канала заняты) в S2. будет иметь интенсивность  $3\mu$ , т.е. может освободиться любой из трех каналов и т.д. [85, 86]

В формуле (9) для n-канальной СМО с отказами получим для предельной вероятности состояния:

$$P_0 = \left( 1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2!\mu^2} + \dots + \frac{\lambda^k}{k!\mu^k} + \dots + \frac{\lambda^n}{n!\mu^n} \right)^{-1}, \quad (9)$$

где члены разложения  $\frac{\lambda}{\mu}, \frac{\lambda^2}{2!\mu^2}, \dots, \frac{\lambda^n}{n!\mu^n}$  будут представлять собой коэффициенты при  $P_0$  в выражениях для предельных вероятностей  $P_1, P_2, \dots, P_k, \dots, P_n$ . Величина  $\rho$  называется приведенной интенсивностью потока заявок или интенсивностью нагрузки канала (10).

$$\rho = \frac{\lambda}{\mu} \quad (10)$$

Она выражает среднее число заявок, приходящее за среднее время обслуживания одной заявки. Формулы (25) и (26) для предельных вероятностей получили названия формул Эрланга в честь основателя теории массового обслуживания.

$$P_1 = \rho \cdot P_0 \quad (11)$$

$$P_2 = \frac{\rho^2}{2!} P_0, \dots, P_k = \frac{\rho^k}{k!} P_0, \dots, P_n = \frac{\rho^n}{n!} P_0. \quad (12)$$

Вероятность отказа СМО – предельная вероятность того, что все n каналов системы будут заняты (13):

$$P_{\text{отж.}} = \frac{\rho^n}{n!} P_0. \quad (13)$$

Относительная пропускная способность – вероятность того, что заявка будет обслужена (14):

$$Q = 1 - P_{\text{ошибка}} = 1 - \frac{\rho^n}{n!} P_0. \quad (14)$$

Абсолютная пропускная способность (15):

$$A = \lambda Q = \lambda \left( 1 - \frac{\rho^n}{n!} P_0 \right). \quad (15)$$

### 3.2.2 Расчёт основных показателей аналитической модели

Для расчёта аналитической модели нам необходимо знать показатели  $\lambda$  и  $\mu$ . Поэтому будем считать, что объекты на транспортной ленте оказываются друг за другом. Следовательно, камера передаёт информацию каждый момент времени (видео поток), но микроконтроллер использует для распознавания только часть получаемых кадров. Интенсивность извлечения (fps) таких кадров и будет  $\lambda$  (16):

$$\lambda = fps. \quad (16)$$

Интенсивность обслуживания, в случае одноканальной СМО с отказами можно рассчитать, зная объём информации и скорость её обработки. Допустим, что каждое изображение состоит из 2073600 пикселей (разрешение изображения – 1920x1080), которые необходимо обойти скользящим окном алгоритма распознавания. Скорость перемещения окна – 6000000 пикселей в секунду. Значит (17):

$$\mu = \frac{6000000}{2073600}. \quad (17)$$

Данные параметры описывают ситуацию, когда вычислительная платформа целиком принимает изображение от камеры и распознаёт его. В случае иерархического подхода с одной итерацией интенсивность обслуживания будет представлена двумя числами. Одна итерация означает, что система предварительно получает и анализирует изображение, сжатое в несколько раз. После обнаружения на сжатом изображении чего-то напоминающего объект, на изображении выделяется область ROI. Данная область получается из полного изображения, но меньше по площади.

Первая строчка (табл. 3.1) описывает систему, работающую с применением стандартных методов. В следующих строчках система, применяя иерархический метод. Допустим, что система изначально анализирует сжатое изображение, площадь которого составляет 82944 пикселя. Для усреднения полученных результатов, предположим, что область ROI всегда в  $n$  раз меньше оригинального изображения. Это позволяет системе обрабатывать  $n$  изображений одновременно, распараллеливая процесс распознавания. Для полноты результатов промоделируем случаи, когда область ROI меньше исходного изображения в 2, 3 и 4 раза (табл. 3.1).

Таблица 3.1

Таблица расчёта интенсивности обслуживания

Площадь ROI	Площадь изображения в пикселях	Скорость обработки (п/с)	Интенсивность обслуживания (1/с)
-	2073600	6000000	2,89
$\frac{1}{2}$	1119744 (1036800+82944)	6000000	5,36
$\frac{1}{3}$	774144 (691200+82944)	6000000	7,75
$\frac{1}{4}$	601344 (518400+82944)	6000000	9,98

Используя эту информацию, рассчитаем с помощью системы компьютерной алгебры из класса систем автоматизированного проектирования – Mathcad 15 – основные параметры СМО. Пример листинга расчёта представлен в приложении Б.

Исходя из полученных данных, можно построить график изменения относительной пропускной способности (рис. 3.4). На нём видно, что Наибольший прирост производительности наблюдается при fps равно 11 при переходе от стандартного метода к иерархическому с ROI равной  $1/2$  площади оригинального изображения (на 24,7%). Однако относительная пропускная

способность равная 98,7% означает, что система распознавания всё ещё иногда будет испытывать перегрузку. В эти моменты времени распознавания будет происходить дольше, чем обычно.

Стоит также отметить, что отказы в реальной системе не означали бы пропуск объекта. В эти моменты времени коэффициент использования превышает 0,7. Реальная система в таком случае увеличит время обслуживания всех заявок в текущий момент до возвращения в обычные условия. Однако функция изменения времени обработки информации нелинейна и труднопрогнозируема из-за зависимости от множества факторов. Поэтому в данной модели для упрощения данные моменты не рассматриваются.

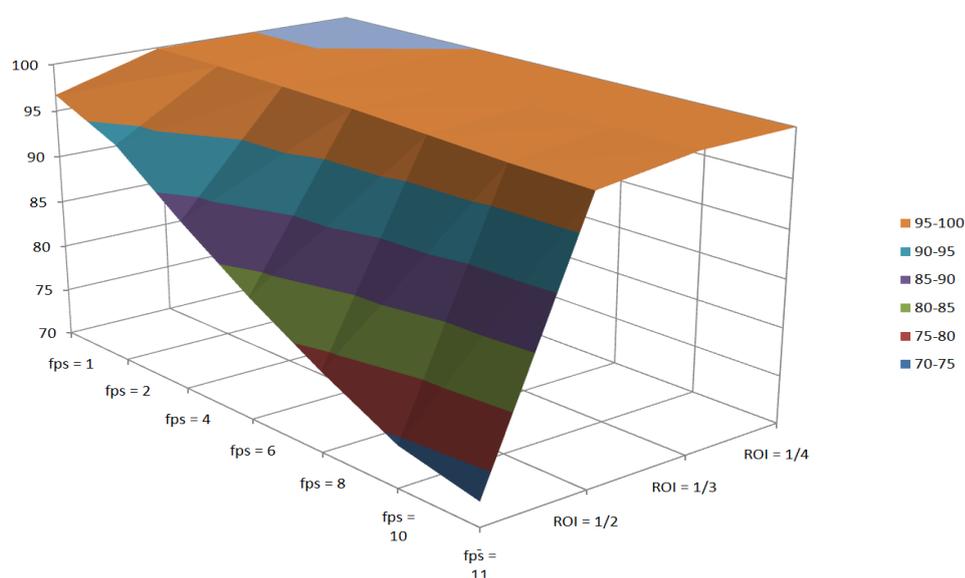


Рисунок 3.4 – относительная пропускная способность СМО

### 3.3 Имитационная модель процесса иерархической обработки растровых изображений

#### 3.3.1 Описание процесса создания имитационной модели

Была разработана имитационная модель на базе программного обеспечения AnyLogic 8 (рисунок 3.5).

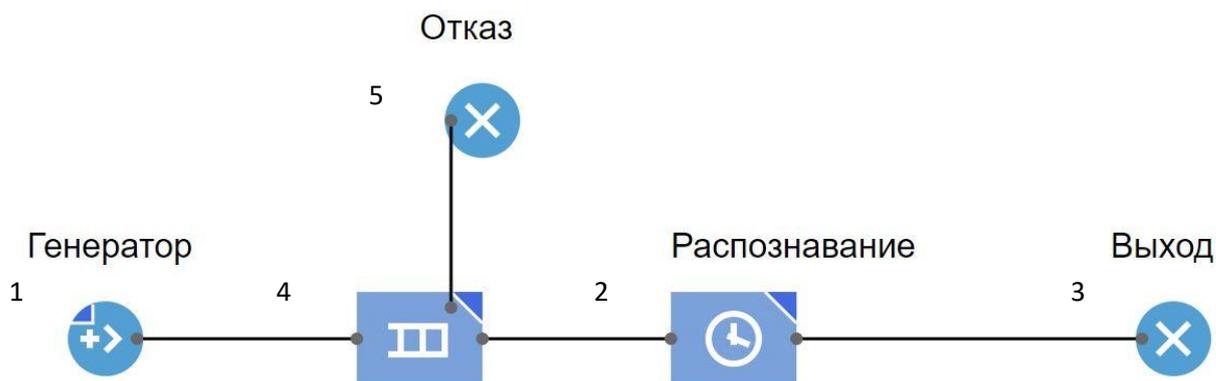


Рисунок 3.5 – Графическое представление имитационной модели

Элемент «1» (блок Source из библиотеки моделирования AnyLogic 8) – рассчитывает генерацию новых заявок (объектов) в системе. Входным параметром для него служит интенсивность прибытия новых заявок, равная 3. А также настроено максимальное количество заявок генерируемых, равное 10000, для упрощения анализа получаемых результатов.

Элемент «2» (блок Delay из библиотеки моделирования AnyLogic 8) – имитирует работу сервера, распознающего объекты. Он принимает и задерживает в себе заявку на время её обслуживания. Другим входным параметром элемента «2» является количество одновременно обрабатываемых заявок. После истечения времени обработки заявка учитывается элементом «3» (блок Sink из библиотеки моделирования AnyLogic 8).

Элемент «4» (блок Queue из библиотеки моделирования AnyLogic 8) – моделирует очередь из объектов, которым не хватило ресурсов для обработки сервером в принятых условиях. Это допущение необходимо чтобы учесть количество возможных случаев перегрузки вычислительной платформы. Заявка находится в очереди в течение 0 секунд. Иначе говоря, все заявки, которым не хватило «вычислительно мощности платформы» получают отказ в обслуживании и учитываются блоком «5» (блок Sink из библиотеки моделирования AnyLogic 8).

### 3.3.2 Расчёт основных показателей имитационной модели

Для расчёта имитационной модели нам необходимо знать показатели  $\lambda$  и время обслуживания заявки ( $t$ ). Поэтому будем считать, что объекты на транспортной ленте оказываются в кадре камеры с интенсивностью 10 штук в минуту, как и при расчёте аналитической модели (15).

Время обслуживания, в случае одноканальной СМО с отказами можно рассчитать, зная объём информации и скорость её обработки. Допустим, что, как и при расчёте аналитической модели, каждое изображение состоит из 2073600 пикселей, которые необходимо обойти скользящим окном алгоритма распознавания. Скорость перемещения окна – 6000000 пикселей в секунду. Значит (18):

$$t = \frac{2073600}{6000000}. \quad (18)$$

Данные параметры описывают ситуацию, когда вычислительная платформа целиком принимает изображение от камеры и распознаёт его. В случае иерархического подхода с одной итерацией интенсивность обслуживания будет представлена двумя числами. Одна итерация означает, что система предварительно получает и анализирует изображение, сжатое в несколько раз. После обнаружения на сжатом изображении чего-то напоминающего объект, на изображении выделяется область ROI. Данная область получается из полного изображения, но меньше по площади.

Первая строчка (табл. 3.2) описывает систему, работающую с применением стандартных методов. В следующих строчках система, применяя иерархический метод. Допустим, что система изначально анализирует изображение, сжатое в 5 раз, т.е. площадь изображения составляет 300 пикселей. Для усреднения полученных результатов, предположим, что область ROI всегда в  $n$  раз меньше оригинального изображения. Это позволяет системе обрабатывать  $n$  изображений одновременно, распараллеливая процесс распознавания. Для полноты результатов

промоделируем случаи, когда область ROI меньше исходного изображения в 2, 3 и 4 раза (табл. 3.2).

Таблица 3.2

Таблица расчёта времени обслуживания

Площадь ROI	Площадь изображения в пикселях	Скорость обработки (п/с)	Время обслуживания (с)
-	2073600	6000000	0,35
$\frac{1}{2}$	1119744 (1036800+82944)	6000000	0,19
$\frac{1}{3}$	774144 (691200+82944)	6000000	0,13
$\frac{1}{4}$	601344 (518400+82944)	6000000	0,10

Область интереса равная  $1/4$  при всех исследуемых режимах в ходе аналитического моделирования показала максимально приближенные к 100% результаты относительной пропускной способности (т.к. вероятность отказа, согласно теории массового обслуживания, не может быть равна 0), поэтому она была проверена имитационным моделированием

Используя эту информацию, рассчитаем с помощью системы компьютерного моделирования – AnyLogic 8 – основные параметры СМО. Примеры результатов имитационного моделирования представлены на рисунке 3.6:

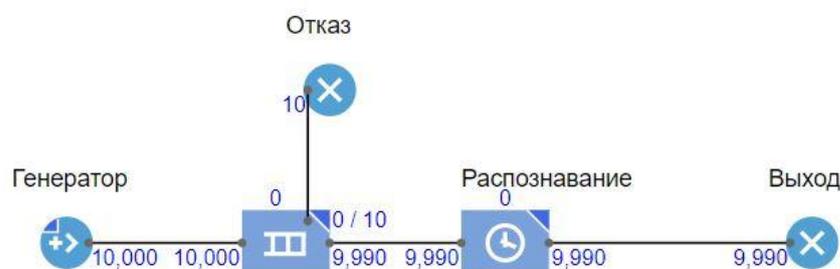


Рисунок 3.6 – Результаты имитационного моделирования (четырёхканальная СМО, одновременно обрабатывающая 4 ROI площадью в  $\frac{1}{4}$  от исходного размера, поступающие с частотой 6 fps)

Исходя из полученных данных, можно построить график изменения относительной пропускной способности (рис. 3.7):

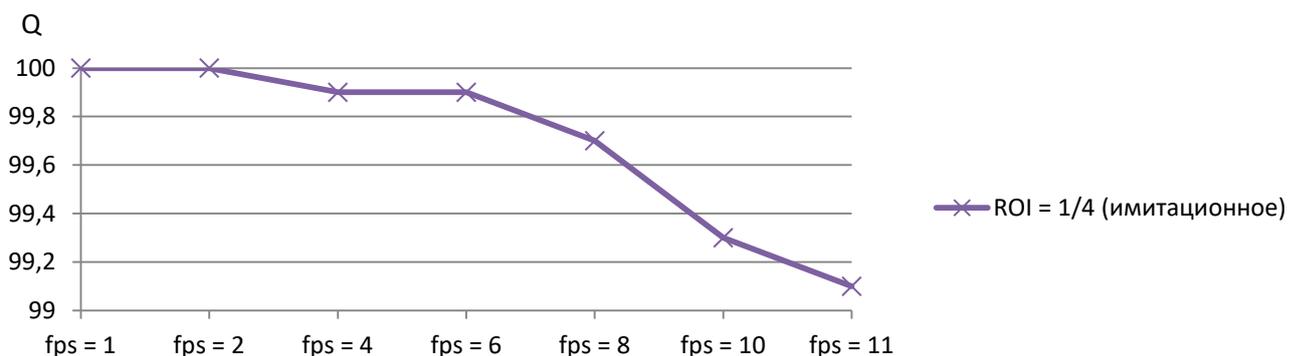


Рисунок 3.7 – относительная пропускная способность СМО

На графике (рис. 3.7) видно, что применение иерархического метода (при первичном сжатии в 5 раз и области ROI площадью в  $\frac{1}{4}$  от исходного размера) позволяет стабильно распознавать все объекты, попадающие в кадр камеры, почти на всём исследуемом интервале fps.

Стоит также отметить, что отказы в реальной системе не означали бы пропуск объекта. В эти моменты времени коэффициент использования превышает 0,7. Реальная система в таком случае увеличит время обслуживания всех заявок в текущий момент до возвращения в обычные условия. Однако функция изменения времени обработки информации нелинейна и труднопрогнозируема из-за зависимости от множества факторов. Поэтому в данной модели для упрощения данные моменты не рассматриваются.

### 3.4 Анализ результатов аналитического и имитационного моделирования

Результаты, полученные в ходе аналитического и имитационного моделирования, представлены на рис. 3.8.

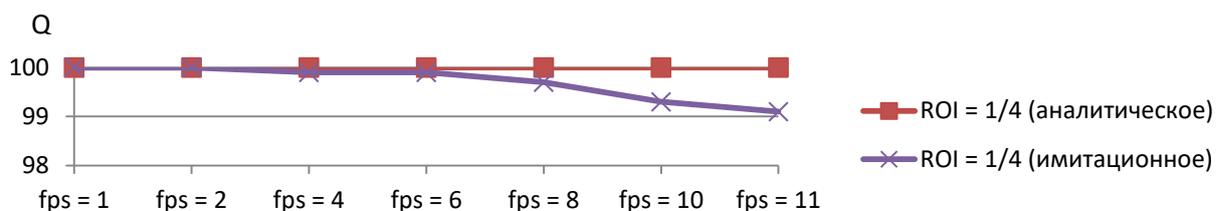


Рисунок 3.8 – Сопоставление полученных значений относительной пропускной способности СМО в ходе аналитического и имитационного моделирования

Для проверки адекватности моделей был выбран критерий Фишера, который позволяет сравнивать величины выборочных дисперсий двух рядов наблюдений. Расчётное значение критерия находится по формуле (19)

$$F_{\text{расчётное}} = \frac{\sum(Y_{i,\text{расчётное}} - Y_{\text{среднее, расчётное}})^2}{m} * \frac{n-m-1}{\sum(Y_i - Y_{i,\text{расчётное}})^2}, \quad (19)$$

где  $Y_{i, \text{ расчётное}}$  – значения, описывающие изменения целевого показателя проверяемой модели,  $Y_{\text{среднее, расчётное}}$  – среднее значение, описывающее изменения целевого показателя проверяемой модели,  $Y_i$  – значения, описывающие изменения целевого показателя основной модели,  $n$  – количество наблюдений,  $m$  – количество факторов, влияющих на степени свободы. Если  $F_{\text{расчётное}}$  будет больше  $F_{\text{табличное}}$ , то исследуемые модели можно считать адекватными. Результаты расчётов приведены в таблице 3.3. На основании полученных результатов можно считать, приведённые модели адекватны.

Влияние на относительную пропускную способность СМО преимущественно оказывает размер обрабатываемого изображения. Так при распознавании полного изображения – алгоритму требует обойти все  $N$  пикселей изображения. При иерархическом распознавании алгоритм вначале обходит сжатое изображение, а затем выбранную область ROI, что суммарно (в большинстве случаев) меньше, чем при обходе полного изображения. Стоит также отметить, что в моменты, когда на транспортной ленте нет объектов, при иерархическом методе система анализирует только сжатое изображение, а при стандартном – всё полное изображение (табл. 3.4).

## Результаты расчётов критерия Фишера

Площадь ROI	$F_{\text{расчётное}}$	$F_{\text{табличное}}$ (для точности 95%)
$\frac{1}{4}$	280,1520	2,4469

Таблица 3.5

## Сопоставление размеров обрабатываемого изображения

	Количество пикселей, подвергаемых обработке, когда объекта нет в кадре	Количество пикселей, подвергаемых обработке, когда объект в кадре
Размер изображения при стандартном методе распознавания	2073600	2073600
Размер изображения при иерархическом методе распознавания (предварительное сжатие, а ROI $\frac{1}{2}$ полного изображения)	82944	1119744
Размер изображения при иерархическом методе распознавания (предварительное сжатие, а ROI $\frac{1}{3}$ полного изображения)	82944	774144
Размер изображения при иерархическом методе распознавания (предварительное сжатие, а ROI $\frac{1}{4}$ полного изображения)	82944	601344

Справедливо утверждение, что меньший объём данных будет передаваться между источником заявки (камерой) и обработчиком заявки (сервером) быстрее. Поэтому данные, полученные в ходе моделирования могут количественно не соответствовать с реальным, так как в модели было принято считать время передачи данных бесконечно малым. Время передачи индивидуально для каждой

системы и зависит от используемых технологий. Чем больше времени занимает процесс передачи информации от источника заявки (камеры) до обработчика заявки (сервера), тем больше будет заметен прирост скорости распознавания изображения.

### 3.6 Выводы по главе

Предложена концепция модели для проведения аналитического и имитационного моделирования. Так условная система для моделирования состоит из камеры (источника заявок) снимающих полотно транспортной ленты, по которой движутся объекты, нуждающиеся в распознавании. Подразумевается, что вычислительная платформа выбрана таким образом, чтобы коэффициент использования был ниже 0,7 в момент 1 полного изображения, получаемого с камеры (источника заявок). Т.е. система всегда работает без перегрузки и колебаниями времени необходимого на выполнение операций можно пренебречь. Считается, что вычислительная платформа не выполняет более никаких функций и расходует вычислительные ресурсы только на процесс распознавания. Время передачи информации от камер (источников заявок) считается бесконечно малым.

Проведено аналитическое и имитационное моделирование заданной системы с помощью программных продуктов Mathcad 15 и AnyLogic 8 соответственно. Результаты показали, что применение иерархического метода (при первичном сжатии и области ROI площадью в  $\frac{1}{4}$  от исходного размера) позволяет значительно повысить скорость распознавания изображений системой по сравнению со стандартным.

Влияние на относительную пропускную способность системы в данном случае преимущественно оказывает размер обрабатываемого изображения. Так при распознавании полного изображения – алгоритму требуется обойти все N пикселей изображения. При иерархическом распознавании алгоритм вначале

обходит сжатое изображение, а затем выбранную область ROI, что суммарно (в большинстве случаев) меньше, чем при обходе полного изображения. Стоит также отметить, что в моменты, когда на транспортной ленте нет объектов, при иерархическом методе система анализирует только сжатое изображение, а при стандартном – всё полное изображение.

## 4. Внедрение иерархического метода распознавания в АСУТП по сортировке мусора

### 4.1 Реализация прототипа системы иерархического распознавания растровых изображений

Реализация прототипа системы проходила на автомате по сортировке твёрдых бытовых отходов «Sortomat» (рисунок 4.1). Как уже отмечалось во второй главе, большинство аналогичных машин используют контроль материала контейнера с помощью ИК-спектрометра, контроль формы контейнера, контроль штрих-кода для проверки тары. В проекте «Sortomat» был применён другой метод идентификации, основанный на распознавании объектов с нейронными сетями, используя контроллер IoT (Raspberry PI 3), что значительно снижает себестоимость автомата, но создаёт ряд трудностей связанных с процессами обработки изображения. [32]



Рис. 4.1 «Sortomat» в сопоставлении с моделью среднестатистического человека

«Sortomat» состоит из контейнера, в который помещаются предметы, поворотного механизма для этого контейнера и сминателя. Над контейнером расположены две видеокамеры. Для полного визуального захвата всего контейнера необходимы две камеры. При использовании только одной камеры требовалось расположить её значительно выше над поверхностью контейнера, что значительно увеличивало высоту автомата. Это означает, что Raspberry PI 3 требуется обрабатывать два изображения, что повышает нагрузку на вычислительную платформу.

Следующей особенностью является возможность принимать не все контейнеры (из-за особенностей технологического процесса утилизации бытовых отходов в регионе принимаются бутылки только из-под воды и газировки, не принимаются бутылки из-под масла, шампуней, молока, грязные бутылки, бутылки, полностью заклеенные этикеткой). В общей сложности количество категорий объектов равно 8, но для классификации эти предметы группируются только в два класса: первый класс – бутылки из-под воды и газировки, второй класс – прочие предметы. Данная задача рационально может быть решена только с помощью нейронных сетей. Это обусловлено тем, что все бутылки имеют разную форму, могут быть деформированы и находится в отсеке под разным углом к оси расположения камер. Raspberry PI 3 (включающий в себя 64-битный четырехъядерный процессор ARMv8 Cortex-A53 с тактовой частотой 1.4 ГГц, модуль оперативной памяти LPDDR2 SDRAM на 512 МБ) позволяет запускать нейронные сети, но из-за большого размера изображения выполняет распознавание медленно (около 1 секунды). Снижение разрешения изображения недопустимо т.к. задача подразумевает возможность различать пластиковые бутылки из-под воды и молока между собой, что реализуется по косвенным признакам (этикетка, форма горлышка и прочее). Кроме того, есть дополнительная задача по поиску штрихкода на упаковке, что при снижении разрешения может сделать его нечитаемым. Стандартные варианты оптимизации работы нейронной сети дают незначительные результаты. [34, 39, 79-83]

Таким образом, в проекте «Sortomat» имеются две взаимосвязанные проблемы, которые необходимо решить:

- 1) Реализация обработки изображения высокого разрешения в условиях ограниченных вычислительных ресурсов;
- 2) Повышение быстродействия нейронной сети в условиях ограниченных вычислительных ресурсов.

Применение иерархического подхода позволяет без потери требуемого функционала снизить максимальное разрешение обрабатываемых изображений. В этом случае нейронная сеть будет задействовать меньшее число вычислительных ресурсов (что положительно скажется на быстродействии системы). Меньшее количество информации тоже позволит получить небольшой прирост в скорости работы самой нейронной сети (требуется меньше операций для обработки меньшего количества информации).

#### **4.2 Тестирование системы иерархического хранения и обработки растровых изображений**

Для проверки предположения, что иерархический метод позволяет повысить скорость обработки, был проведён эксперимент. Нейронная сеть MobileNet [44], обученная на поиск ПЭТ [45, 56-58б 63] бутылок была запущена на трёх видеофрагментах одинакового содержания (пример кадра представлен на рисунке 4.2), но разных по разрешению:

- 1) изображение размером 1920x1080 пикселей (30 fps) имитирующее видеопоток, полученный от камеры сортирующего автомата;
- 2) изображение размером 192x108 пикселей (сжатое в 10 раз первое видео) (30 fps) имитирующее видеопоток, подготовленный для поиска ROI с помощью SmartCamera;
- 3) изображение 500x1080 (вырезанные фрагменты изображения первого видео, на котором были обнаружены объекты отличные от сцены

распознавания) (30 fps) имитирующие отправленные на распознавание ROI.



Рисунок 4.2 – Пример кадра видео, используемого для теста

Сеть MobileNet была выбрана из-за того, что она позволяет динамически изменять разрешение изображения принимаемого на входе. Это позволяет гарантировать, что при перенастройке нейронной сети не возникнет непредвиденных изменений параметров, способных оказать влияние на скорость выполнения каждой отдельной операции распознавания. Другими словами – скорость распознавания входящего изображения будет зависеть только от алгоритма и разрешения самого изображения.

Все три видео были проанализированы нейронной сетью, которая воспринимает каждый пятый кадр видеопотока как отдельное изображение (это обусловлено отсутствием необходимости фиксирования очень быстрых движений объекта в кадре). После каждых 50 изображений производились замеры времени обработки фрагмента. Подобных замеров было произведено 5 штук – необходимо для того чтобы минимизировать возможные погрешности результатов, связанные с особенностями работы используемого компьютера и программного обеспечения.

Результаты распознаваний в данном случае не важны, так как нейронной сети при любом её вердикте приходится производить функцию свёртки для всего изображения. Следовательно, время выполнения зависит только размера распознаваемого изображения и количества доступных вычислительных ресурсов в конкретный момент времени.

Результаты замеров представлены в таблице 4.1, на рисунке 4.3 и 4.4. Время, затраченное на анализ первого видеопотока, представлено в столбце «Анализ полного изображения». В столбце «Поиск ROI» представлено время распознавания второго видеопотока. В столбце «Иерархический анализ» – сумма времени распознавания второго и третьего – полное время распознавания иерархическим методом (без учёта времени необходимого для сохранения и считывания изображения в память компьютера). Такое допущение позволительно, так как время операций над изображением в оперативной памяти будет оказывать небольшое влияние на конечную величину по сравнению с самим временем распознавания. Время необходимое для передачи изображения серверу тоже не будет учитываться для упрощения.

Таблица 4.1

## Результаты эксперимента

Номер эксперимента	Анализ полного изображения (с)	Поиск ROI (с)	Иерархический анализ (с)
1 (50 изображений)	21,0045	0,6956	7,8712
2 (50 изображений)	20,3127	0,6399	7,9956
3 (50 изображений)	20,9035	0,6001	6,6439
4 (50 изображений)	20,5047	0,5476	6,1611
5 (50 изображений)	21,6109	0,643	7,7549
Среднее значение (50 изображений)	20,8673	0,6252	7,2853
Среднее значение (1 изображение)	0,4173	0,0125	0,1457

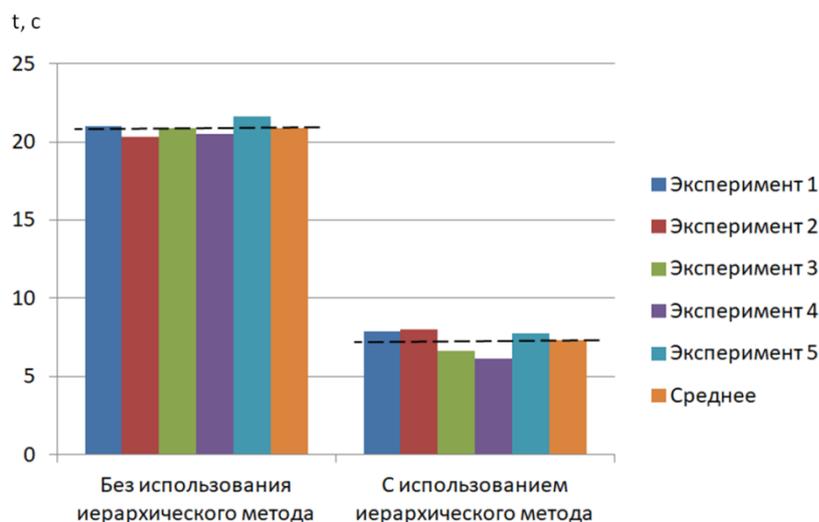


Рис. 4.3. График экспериментального времени для распознавания одного кадра с применением выбранного иерархического метода и без него

Соответственно среднее экспериментальное время распознавания одного кадра иерархическим методом заняло 0,1551 секунды, а стандартным – 0,4322 секунды (рис. 4.5). Разность между ними составила 0,2771 секунды (на 64% быстрее), что подтверждает расчёты, представленным во второй главе, Согласно им процесс распознавание иерархическим методом в заданных условиях быстрее распознавания стандартным методом на 62%. Адекватность полученных результатов по критерию Фишера – выше 95%.

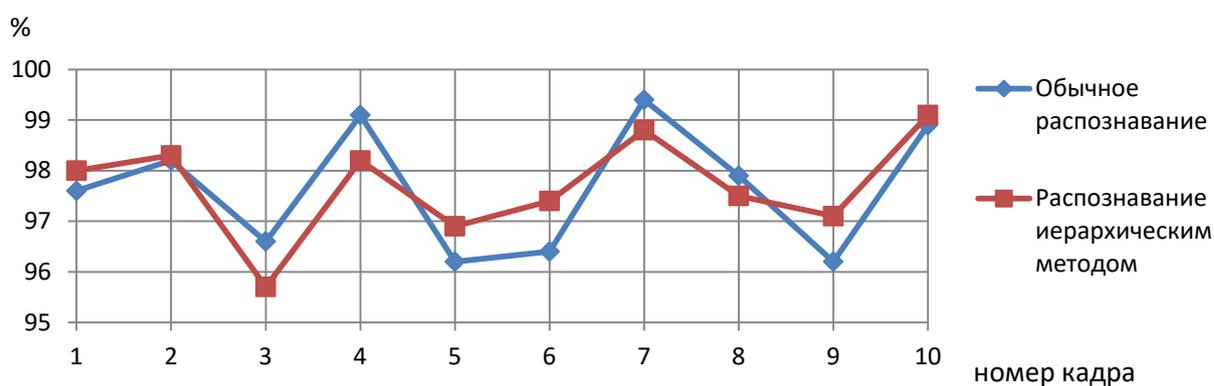


Рис. 4.4. График правильности распознавания объекта на кадре

t, c

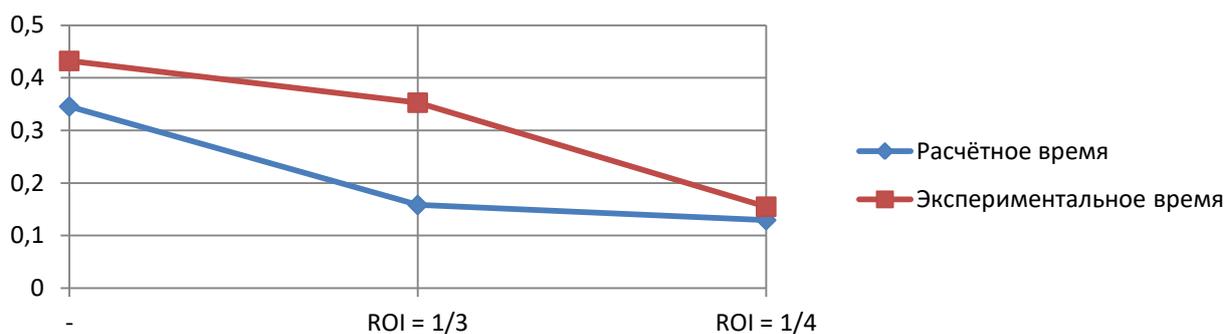


Рис. 4.5. График времени распознавания объекта на кадре

Разница в показателях времени распознавания между расчётным и экспериментальным временем, прежде всего, обусловлена наличием дополнительных временных задержек в работе реальной системы, вызванных записью и чтением информации из памяти контролера, а также техническими особенностями работы операционной системы. Экспериментальные результаты, полученные на том же автомате, но при  $ROI = 1/3$ , демонстрируют влияние загруженности системы на скорость обработки данных – наблюдается уменьшение времени распознавания, однако в меньшей степени из-за периодической нехватки вычислительных ресурсов. Таким образом, разработанные в третьей главе, модели позволяют рассчитать параметры системы, при которых иерархический алгоритм распознавания может быть применён эффективно.

Подобные расчёты могут быть выполнены для любых систем, занимающихся визуальным распознаванием объекта на изображении. При больших вычислительных возможностях платформы и/или меньшей вычислительной сложности алгоритмов распознавания (при заданном размере изображения) различия между временными показателями обычного и иерархического распознавания будут меньше. При значительной начальной нехватке вычислительных ресурсов – рост производительности будет наглядней, но может не достичь оптимальных значений из-за разумного ограничения размера ROI, всё ещё позволяющего распознавать объекты с заданной достоверностью.

### 4.3 Внедрение результатов в проекте RVM «Sortomat»

Для внедрения иерархического метода распознавания в RVM «Sortomat» было необходимо произвести изменения в конструкции и программном обеспечении автомата.

Была разработана программа бинокулярного зрения (приложение Е). Она реализует процесс склейки изображений с двух камер с учетом расстояния до объекта. Программа выполняет алгоритм сопоставления двух изображений для обнаружения «общего края». Это позволяет получать изображения предмета при его близком расположении к камерам без смещений, вызванных смещением объекта в кадре (из-за разного фокусного расстояния камер, эффекта параллакса, дефектов формы распознаваемого объекта и т.д.) (приложение Ж). Алгоритм опирается на сведения, получаемые от матриц камер и датчика расстояния, установленного в RVM «Sortomat». Алгоритм работы самого автомата приобретает следующий вид (рисунок 4.6):

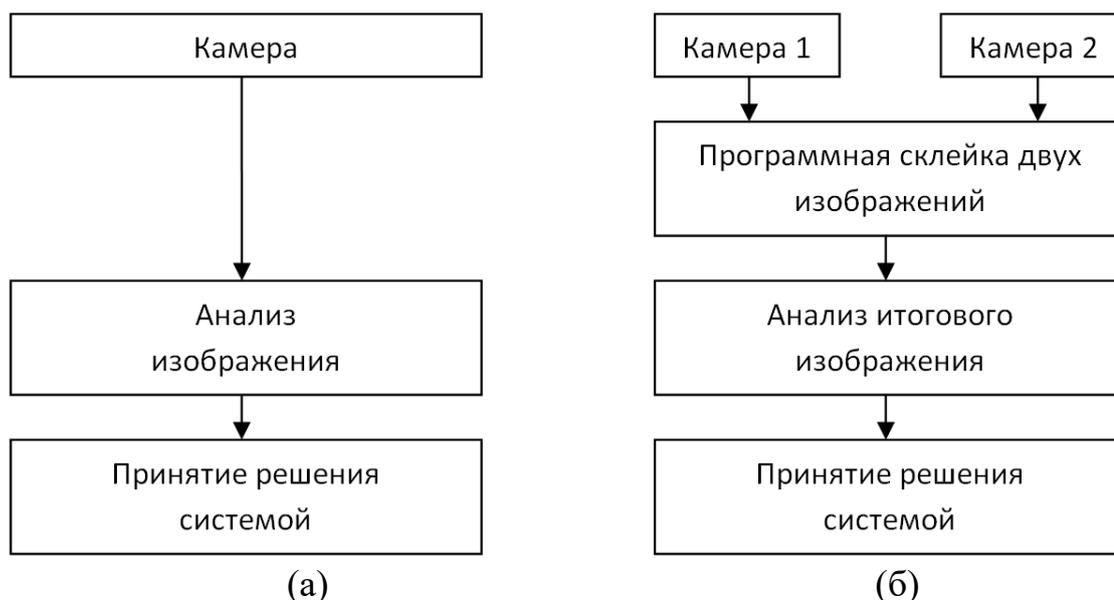


Рисунок 4.6 – Схема работы автомата по приёму тары (а – классической компоновки, б – предлагаемой компоновки)

Для внесения изменений в конструкцию автомата была разработана полезная модель (приложение 3). Она учитывает возможность установки двух камер соосно отсеку распознавания, как показано на рисунке 4.7.

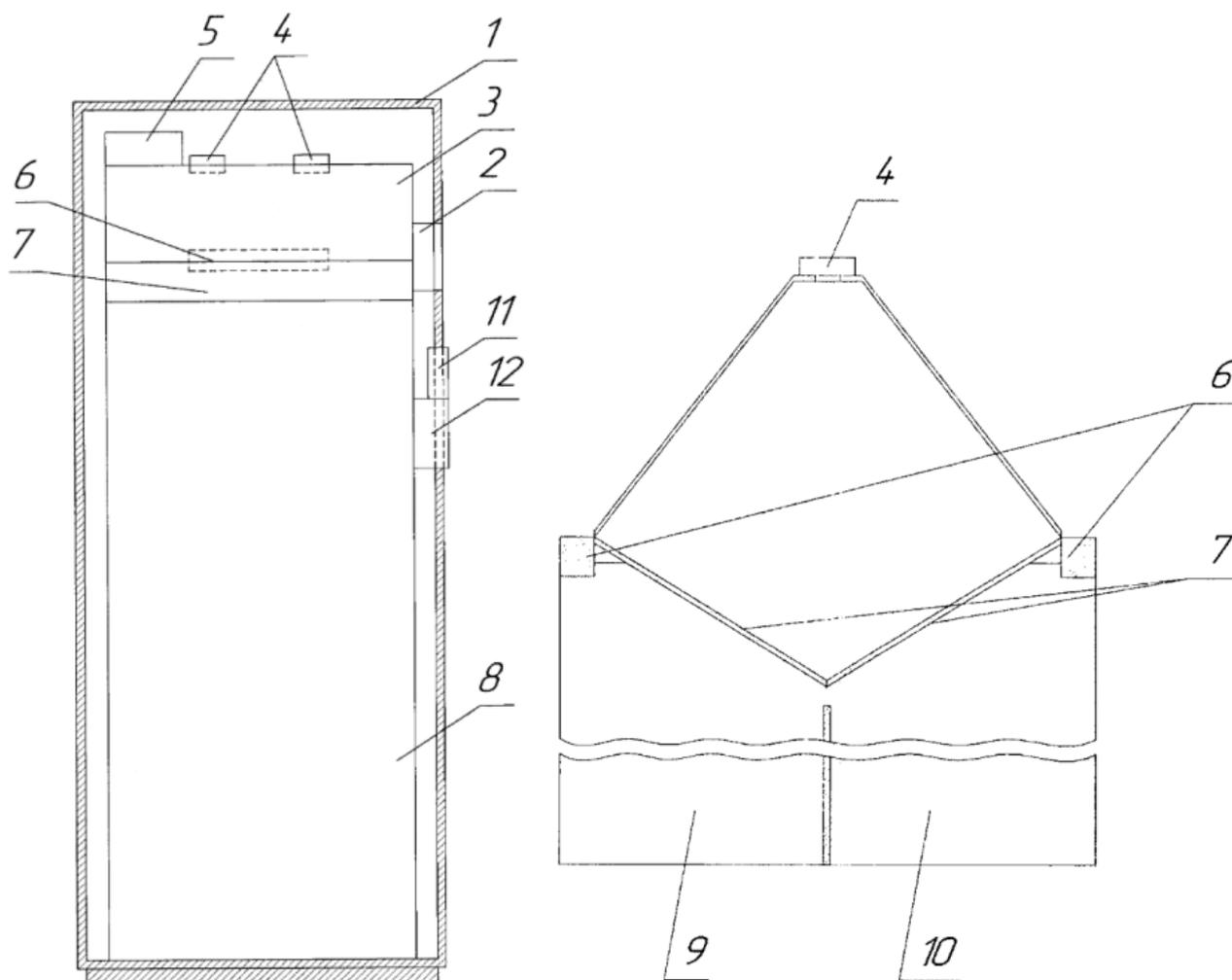


Рисунок 4.7 – Схема расположения элементов в модернизированной конструкции RVM «Sortomat»

На рисунке 4.3 цифрой 1 обозначен корпус автомата, 2 – приёмное отверстие, 3 – отсек распознавания, 4 – видеокамеры, 5 – блок управления автоматом (микроконтроллер). 6 – запирающие устройства (электромагнитные толкатели), 7 – сортирующие створки. В нижней части автомата расположен отсек хранения принятой тары – 8, разделённый перегородкой на две части 9 и 10. На лицевой панели автомат оснащён дисплеем – 11 и средством выдачи вознаграждения – 12.

Подобная конструкция позволяет уменьшить высоту отсека распознавания до 50%, что открывает возможность для увеличения полезного объёма отсека хранения. Это снижает его потенциальный срок самоокупаемости и увеличивает время, которое RVM может работать без обслуживания, что подтверждено актами внедрения в приложениях Е и Ж.

Применение широкоугольных объективов не желательно, т.к. они в значительной степени искажают края, что негативно влияет на распознавание. Этот недостаток можно устранить, увеличив расстояние между камерой и рабочей поверхностью отсека распознавания так, чтобы контейнер не оказывался в искажаемых областях изображения. Однако это решение будет уступать по получаемой выгоде выбранному (рисунок 4.7).

Установка большего числа камер повышает техническую сложность реализации автомата и даёт незначительное уменьшение высоты отсека распознавания из-за ограничения по фокусному расстоянию.

Таким образом, внедрение иерархического подхода в проект RVM «Sortomat» позволяет решить следующий ряд проблем:

- Расширение функционала автомата (возможность распознавать штрихкоды без потери качества распознавания тары);
- Повышение полезного объёма отсека хранения без потери качества распознавания тары;
- Создание модификаций автомата для отсеков распознавания с габаритами отличными от оригинального.

#### **4.4 Выводы по главе**

Реализация прототипа системы проходила на автомате по сортировке твёрдых бытовых отходов «Sortomat» в котором был применён метод идентификации, основанный на распознавании объектов с нейронными сетями,

используя контроллер IoT (Raspberry PI 3). Применение иерархического подхода позволило без потери требуемого функционала снизить максимальное разрешение обрабатываемых изображений, то есть нейронная сеть будет задействовать меньшее число вычислительных ресурсов (что положительно скажется на быстродействии системы), а меньшее количество информации позволит быстрее выполнять весь цикл алгоритма распознавания. Согласно результатам проведённого эксперимента среднее время распознавания видеопотока при стандартном подходе на 1,1820 секунды медленнее распознавания иерархическим методом. Это означает, что сервер в среднем, при прочих равных и в условиях достаточного количества вычислительных ресурсов, будет выполнять те же функции, но на 31,9% быстрее.

## Заключение

Был проведён анализ существующих систем распознавания информации, представленной в графическом виде, с целью выявления возможности оптимизации использования вычислительных ресурсов. В результате было определено, что наиболее выделяются среди прочих алгоритмы, основанные на использовании сверточных нейронных сетей. Они демонстрируют выдающееся качество в решении задачи распознавания, но главным препятствием их применения в системах является скорость работы, которой недостаточно для обработки больших объёмов информации. Для повышения производительности таких систем предложено использовать иерархический подход, основанный на подходе «Coarse-to-Fine».

Предложенный метод обработки данных (иерархический метод распознавания), представленный в графическом виде, имеет преимущество над стандартными по времени распознавания исходного изображения и по ресурсозатратности так как позволяет значительно уменьшить разрешение обрабатываемого изображения, что снижает общее время обработки.

Для моделирования процесса распознавания иерархическим методом была предложена условная система, состоящая из камеры (источника заявок) снимающих полотно транспортной ленты, по которой движутся объекты, нуждающиеся в распознавании. Для облегчения процесса моделирования также были введены дополнительные условия:

- система всегда работает без перегрузки и колебаниями времени необходимого на выполнение операций можно пренебречь;
- вычислительная платформа не выполняет более никаких функций и расходует вычислительные ресурсы только на процесс распознавания;
- время передачи информации от камер (источников заявок) считается бесконечно малым.

Для описанной системы моделирования производилось в Mathcad 15 (аналитическое моделирование) и AnyLogic 8 (имитационное моделирование). Результаты показали, что применение иерархического метода позволяет значительно повысить скорость распознавания изображений системой по сравнению со стандартным. Влияние на выходные характеристики системы в преимущественно оказал размер обрабатываемого изображения.

Эксперимент по применению иерархического подхода к распознаванию данных, представленных в графическом виде в АСУТП был проведён на прототипе автомата по сортировке твёрдых бытовых отходов «Sortomat» в котором был применён метод идентификации, основанный на распознавании объектов с нейронными сетями, используя контроллер IoT (Raspberry PI 3). Согласно результатам проведённого эксперимента среднее время распознавания видеопотока при стандартном подходе на 0,2771 секунды медленнее распознавания иерархическим методом. Это означает, что сервер в среднем, при прочих равных и в условиях достаточного количества вычислительных ресурсов, будет выполнять те же функции, но на 64% быстрее.

На основании полученных в ходе эксперимента результатов была разработана модификация RVM «Sortomat». Для этого были произведены изменения в конструкции и программном обеспечении автомата. Внедрение иерархического подхода в проект позволило решить следующий ряд проблем:

- Расширение функционала автомата (возможность распознавать штрихкоды без потери качества распознавания тары);
- Повышение полезного объёма отсека хранения без потери качества распознавания тары;
- Создание модификаций автомата для отсеков распознавания с габаритами отличными от оригинального.

## Список сокращений

BSD	Berkeley Software Distribution
Caffe	Convolution Architecture For Feature Extraction
CNN	Convolutional Neural Network
FPS	Frames per Second
HOG	Histogram of Oriented Gradients
IoT	Internet of things
OpenCV	Open Source Computer Vision Library
ROI	Region Of Interest
RVM	Reverse vending machine
SC	SmartCamera
SIFT	Scale-Invariant Feature Transform
АСТПП	Автоматизированная Система Технологической Подготовки Производства
АСУП	Автоматизированная Система Управления Предприятием
АСУТП	Автоматизированная Система Управления Технологическим Процессом
АЦП	Аналого-Цифровой Преобразователь
ПЭТ	Полиэтиленгликольтерефталат
СМО	Система Массового Обслуживания
ТКО	Твёрдые Коммунальные Отходы
ФЗ	Федеральный Закон
ЦОД	Центр Обработки Данных, Дата-Центр
ЦОС	Цифровая Обработка Сигналов

## Список терминов

AlexNet: сверточная нейронная сеть, усложнённый вариант LeNet, увеличенный в тысячу раз.

big data: обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия.

BSD (Berkeley Software Distribution): система распространения программного обеспечения в исходных кодах, созданная для обмена опытом между учебными заведениями. Лицензия BSD допускает проприетарное коммерческое использование ПО. Для ПО, выпущенного под этой лицензией, допускается встраивание в проприетарные коммерческие продукты. Работы, основанные на таком ПО, даже могут распространяться под проприетарными лицензиями (но всё же обязаны соответствовать требованиям лицензии).

Caffe (Convolution Architecture For Feature Extraction): открытое программное обеспечение, среда для глубокого обучения, разработанная Яньцинем Цзя в процессе подготовки своей диссертации в университете Беркли.

cloud computing: модель обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых вычислительных ресурсов (например, сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам – как вместе, так и по отдельности), которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами или обращениями к провайдеру.

CNN (convolutional neural network): свёрточная нейронная сеть специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание образов.

data augmentation: это методика создания дополнительных обучающих данных из имеющихся данных. Существует множество вариантов аугментации: отражение по горизонтали, случайное кадрирование и изменение цвета и другие.

Можно применять различные комбинации вариантов для большего разнообразия выборки.

**DropOut:** метод регуляризации искусственных нейронных сетей, предназначен для уменьшения переобучения сети за счет предотвращения сложных коадаптаций отдельных нейронов на тренировочных данных во время обучения, характеризуется исключением определённого процента случайных нейронов (находящихся как в скрытых, так и видимых слоях) на разных эпохах во время обучения нейронной сети. Такой приём значительно увеличивает скорость обучения, качество обучения на тренировочных данных, а также повышает качество предсказаний модели на новых тестовых данных.

**edge computing:** модель обработки информации, позволяющая распределять вычислительные ресурсы и сервисы приложений по каналу связи через децентрализованную вычислительную инфраструктуру.

**FPS (Frames per Second) :** количество сменяемых кадров за единицу.

**green computing:** модель обработки информации, ориентированная на энергосберегающие, энергоэффективные технологии вычислений.

**HOG (Histogram of Oriented Gradients):** дескрипторы особых точек, которые используются в компьютерном зрении и обработке изображений с целью распознавания объектов. Данная техника основана на подсчете количества направлений градиента в локальных областях изображения. Этот метод похож на гистограммы направления края, дескрипторы SIFT и контексты формы, но отличается тем, что вычисляется на плотной сетке равномерно распределенных ячеек и использует нормализацию перекрывающегося локального контраста для увеличения точности.

**IoT (Internet of things):** концепция вычислительной сети физических предметов, оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключаящее из части действий и операций необходимость участия человека.

LeNet: свёрточная нейронная сеть, представленная в 1998 году французским исследователем Яном Лекуном, базовая модель состоит из трёх типов слоёв: свёрточные (convolutional) слои, субдискретизирующие (subsampling, подвыборка) слои и слои "обычной" нейронной сети – перцептрона.

MapReduce: это модель распределенной обработки данных, предложенная компанией Google для обработки больших объёмов данных на компьютерных кластерах.

matplotlib: кроссплатформенная библиотека для создания 2D графиков из данных в массивах. Предоставляет объектно-ориентированный API, который помогает встраивать графики в приложения, используя наборы инструментов Python GUI. Может использоваться в оболочках Python и IPython, ноутбуках Jupyter и серверах веб-приложений.

MobileNet: свёрточная нейронная сеть, предложенная компанией Google для использования в системах компьютерного зрения на мобильных платформах, направлена на уменьшение количества требуемых вычислительных ресурсов.

numpy: это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

OpenCV (Open Source Computer Vision Library): библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом.

Python: высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода

ROI (Region Of Interest): область изображения на которой находится важная для распознавания информация.

RVM (Reverse vending machine): автомат по приёму тары, позволяющий людям возвращать пустые контейнеры (бутылки ПЭТ и алюминиевые банки) для напитков за вознаграждение.

SC (SmartCamera): это камера, выполненная на базе микроконтроллера IoT, способная проводить предварительную обработку передаваемого серверу изображения и выполнять некоторые логические функции.

SIFT (scale-invariant feature transform ): алгоритмом выявления признаков в компьютерном зрении для выявления и описания локальных признаков в изображениях. Строится на извлечении ключевых точек объектов из набора контрольных изображений. Объект распознаётся в новом изображении путём сравнения каждого признака из нового изображения с признаками из базы данных и нахождения признаков-кандидатов на основе евклидова расстояния между векторами признаков.

SURF: алгоритм поиска ключевых точек изображения и создание их дескрипторов, инвариантных к масштабу и вращению. Метод ищет ключевые точки с помощью матрицы Гессе. Детерминант матрицы Гессе (гессиан) достигает экстремума в точках максимального изменения градиента яркости.

TensorFlow: открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия.

ZBar: это библиотека с открытым исходным кодом для считывания штрихкодов из видеопотока или отдельных изображений.

АЦП (Аналого-цифровой преобразователь): устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал).

гамма излучение: вид электромагнитного излучения, характеризующийся чрезвычайно малой длиной волны — менее  $2 \cdot 10^{-10}$  м — и, вследствие этого, ярко выраженными корпускулярными и слабо выраженными волновыми свойствами. Относится к ионизирующим излучениям, то есть к излучениям, взаимодействие которых с веществом способно приводить к образованию ионов разных знаков.

инфракрасное излучение: электромагнитное излучение, занимающее спектральную область между красным концом видимого света (с длиной волны  $\lambda = 0,74$  мкм и частотой 430 ТГц) и микроволновым радиоизлучением ( $\lambda \sim 1$ —

2 мм, частота 300 ГГц). Оптические свойства веществ в инфракрасном излучении значительно отличаются от их свойств в видимом излучении и зависят, прежде всего, от их химического состава.

нейронные сети (искусственная нейронная сеть): математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов).

ПЭТ (полиэтиленгликольтерефталат, ПЭТФ, ПЭТГ, лавсан, майлар): термопластик, наиболее распространённый представитель класса полиэфиров. Твёрдое, бесцветное, прозрачное вещество в аморфном состоянии и белое, непрозрачное в кристаллическом состоянии. Применяется в создании пищевых контейнеров.

рециклинг: процесс переработки отходов в материал, который можно использовать вторично.

СМО (система массового обслуживания): математический (абстрактный) объект, содержащий один или несколько приборов (каналов), обслуживающих заявки, поступающие в систему.

спектрометр: оптический прибор, предназначенный для фиксации отражённых волн от исследуемого объекта (спектра), их количественной обработки и дальнейшего анализа.

ТКО (твёрдые коммунальные отходы): предметы или товары, потерявшие потребительские свойства, наибольшая часть отходов потребления. ТКО делятся также на (биологические) отбросы и собственно бытовой мусор (небиологические твёрдые отходы искусственного или естественного происхождения).

ЦОД (центр обработки данных, дата-центр): это специализированное здание для размещения серверного и сетевого оборудования и подключения абонентов к каналам сети Интернет. Исполняет функции обработки, хранения и распространения информации. Объединение вычислительных ресурсов и средств

хранения данных в ЦОД позволяет сократить совокупную стоимость владения ИТ-инфраструктурой за счёт возможности эффективного использования технических средств, например, перераспределения нагрузок, а также за счёт сокращения расходов на администрирование.

ЦОС (цифровая обработка сигналов): способы обработки сигналов на основе численных методов с использованием цифровой вычислительной техники.

штрихкод: графическая информация о товаре в виде последовательности штрихов и пробелов различной ширины и нескольких цифр, расположенных под ними.

## Список литературы

1. Akhmetzyanov K., Yuzhakov A. Waste Sorting Neural Network Architecture Optimization, 2019 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 2019, pp. 1-5. doi: 10.1109/RUSAUTOCON.2019.8867749.
2. Alberto Leon-Garcia. Probability, statistics, and random processes for electrical engineering. — 3rd. — Prentice Hall, 2008. — ISBN 0-13-147122-8.
3. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv препринт arXiv:1704.04861, 2017. URL: <https://arxiv.org/pdf/1704.04861.pdf> (дата обращения: 16.12.2018).
4. Anil Bas & William A. P. Smith. What Does 2D Geometric Information Really Tell Us About 3D Face Shape? // International Journal of Computer Vision, volume 127, pages 1455–1473, 2019
5. Bengio Y. Practical Recommendations for Gradient-Based Training of Deep Architectures // Neural Networks: Tricks of the Trade. - 2012. - №7700. С. 437-478. DOI: 10.1007/978-3-642-35289-8\_26.
6. BernathPeter F. Infrared fourier transform emission spectroscopy // Chemical Society Reviews. 1996. Т. 25, вып. 2. С. 111-115. doi: 10.1039/CS9962500111
7. Bhattacharyya A. On a measure of divergence between two statistical populations defined by their probability distributions// Bulletin of the Calcutta Mathematical Society 35: 99–109. MR0010358, 1943.
8. Big Data от А до Я. [Электронный ресурс] // Data-Centric Alliance. URL: [http://datacentric.ru/blog/big\\_data\\_ot\\_a\\_do\\_ya.html](http://datacentric.ru/blog/big_data_ot_a_do_ya.html) (дата обращения 15.05.17)
9. Caffe [Электронный ресурс]. URL: [caffe.berkeleyvision.org](http://caffe.berkeleyvision.org) (дата обращения: 16.12.2018).

10. Chao Chen. Active Burst-Buffer: In-Transit Processing Integrated into Hierarchical Storage // 11th IEEE International Conference on Networking, Architecture, and Storage (NAS'16), 2016
11. Cheng Lei and Yee-Hong Yang. Optical Flow Estimation on Coarse-to-Fine Region-Trees using Discrete Optimization [режим доступа: [https://cs.brown.edu/courses/cs296-4/Papers/2010/iccv2009\\_201.pdf](https://cs.brown.edu/courses/cs296-4/Papers/2010/iccv2009_201.pdf)]
12. Clay D. Spence, John C. Pearson, and Jim Bergen. Coarse-to-Fine Image Search Using Neural Networks [режим доступа: <https://papers.nips.cc/paper/982-coarse-to-fine-image-search-using-neural-networks.pdf>]
13. Collins R. Mean-shift Blob Tracking through Scale Space // Computer Vision and Pattern Recognition (CVPR'03), June 2003.
14. Comaniciu D., Ramesh V., Meer P. Kernel-based object tracking // IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 25, No 5, may 2003.
15. Comaniciu D., Ramesh V., Meer P. Real-Time Tracking of Non-Rigid Objects using Mean Shift // IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, South Carolina. Vol. 2, pp. 142–149, 2000.
16. Cost of a Deposit Return Reverse Vending Machine [Электронный ресурс]. URL: <http://www.zerowastescotland.org.uk/sites/default/files/reverse%20v%20-%20CFE%20response.pdf> (дата обращения 30.12.2017).
17. DalmaMarinus, VuxtonaMichael W. N., van RuitenbeekbFrank J.A., VonckenaJack H.L. Application of near-infrared spectroscopy to sensor based sorting of a porphyry copper ore // Minerals Engineering. 2014. Т. 58. С. 7-16. doi: 10.1016/j.mineng.2013.12.016
18. Daniel Gehrig, Henri Rebecq, Guillermo Gallego & Davide Scaramuzza. EKLT: Asynchronous Photometric Feature Tracking Using Events and Frames // International Journal of Computer Vision, volume 128, pages 601–618, 2020
19. DL4J, Torch7, Theano and Caffe. URL: <https://deeplearning4j.org/compare-dl4j-tensorflow-pytorch> (дата обращения: 15.03.2018)

20. Dmytro Derkach, Adria Ruiz & Federico M. Sukno. Tensor Decomposition and Non-linear Manifold Modeling for 3D Head Pose Estimation // International Journal of Computer Vision, volume 127, pages 1565–1585, 2019
21. El-Amir, Hisham, Hamdy, Mahmoud // Deep Learning Pipeline. 2020, ISBN 978-1-4842-5349-6
22. Everitt B.S. (2002) Cambridge Dictionary of Statistics, CUP. ISBN 0-521-81099-X (entry for «Shrinkage»)
23. Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv препринт arXiv:1602.07360, 2016. URL: <https://arxiv.org/pdf/1602.07360.pdf> (дата обращения: 16.12.2018).
24. Glorot, X. Deep sparse rectifier neural networks / X. Glorot, A. Bordes, Y. Bengio // Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS). – 2011. – P. 315-323.
25. Howard Andrew G., Zhu Menglong, Chen Bo, Kalenichenko Dmitry, Wang Weijun, Weyand Tobias, Andreetto Marco, Adam Hartwig. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv препринт arXiv:1704.04861, 2017.
26. Iandola F. Forrest, Han Song, Moskewicz W. Matthew et. al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 Mb model size, ICLR'17 conference proceedings, p.207-212, 2017.
27. IugaA., MorarR., SamuilaA., DascalescuL. Electrostatic separation of metals and plastics from granular industrial wastes // IEE Proceedings -Science, Measurement and Technology. 2001. Т. 148, вып. 2.С. 47-54. doi: 10.1049/ip-smt:20010356
28. Jiu Huang, Thomas Pretz, Zhengfu Bian. Intelligent solid waste processing using optical sensor based sorting technology // Image and Signal Processing (CISP), 2010 3rd International Congress on. Yantai, 16-18 окт., 2010. Т. 4. С. 1657-1661. doi: 10.1109/CISP.2010.5647729

29. Jia Yangqing, Shelhamer Evan, Donahue Jeff, Karayev Sergey, Long Jonathan, Girshick Ross, Guadarrama Sergio, Darrell Trevor. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv препринт arXiv:1408.5093, 2014.
30. Jian Liu, Hossein Rahmani, Naveed Akhtar & Ajmal Mian. Learning Human Pose Models from Synthesized Data for Robust RGB-D Action Recognition // International Journal of Computer Vision, volume 127, pages 1545–1564, 2019
31. Kokoulin A. N., Tur A. I., Yuzhakov A. A. Convolutional Neural Networks Application in Plastic Waste Recognition and Sorting // Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (2018 ElConRus). Saint-Petersburg: Saint Petersburg Electrotechnical University «LETI». P. 1118–1122.
32. Kokoulin, A.N., Yuzhakov, A.A., Tur, A.I., Knyazev, A.I. The optical method for the plastic waste recognition and sorting in a reverse vending machine // International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, SGEM, 19 (4.1), pp. 793-800.
33. Kornelik. Дроны инспекторы для промышленных предприятий: основные направления и модели [Электронный ресурс]. URL: <http://integral-russia.ru/2018/10/23/drony-inspektory-dlya-promyshlennyh-predpriyatij-osnovnye-napravleniya-i-modeli/> (Дата обращения: 18.02.2018).
34. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // Proceedings of the 25th International Conference on Neural Information Processing Systems. (NIPS'12) / Curran Associates Inc. December 03 – 06, 2012, Lake Tahoe, Nevada, USA; 2012, T. 1, с. 1106–1114.
35. Lasoff M. A. An Rv By Any Other Name // Waste Age – 2000. – Vol 31 №7. – P. 34.
36. Lebedev. Использование памяти - Intel FPGA vs Nvidia GPU [Электронный ресурс]. URL: <https://fpga->

- systems.ru/news/statja\_ispolzovanie\_pamjati\_intel\_fpga\_vs\_nvidia\_gpu/2020-02-20-273. (Дата обращения: 18.02.2018).
37. LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. - 2015. - № 521. - С. 436-444. DOI: 10.1038/nature14539.
  38. Liu, Zhuo; Lofstead, Jay; Wang, Teng; Yu, Weikuan (September 2013). "A Case of System-Wide Power Management for Scientific Applications". 2013 IEEE International Conference on Cluster Computing (CLUSTER). IEEE. pp. 1–8. doi:10.1109/CLUSTER.2013.6702681. ISBN 978-1-4799-0898-1.
  39. Lorraine J., Duvenaud D. Stochastic hyperparameter optimization through hypernetworks. arXiv препринт arXiv:1802.09419, 2018. URL: <https://arxiv.org/pdf/1802.09419.pdf> (дата обращения: 20.09.2019).
  40. Lucas B., Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vi-sion // International Joint Conference on Artificial Intelligence, pp. 674–679, 1981.
  41. Maggio E., Cavallaro A. Video tracking theory and practice. – Wiley, 2011. Szelisky R. Computervision: algorithm sandapplication. – Springer, 2010.
  42. Mesina M.B., de Jong T.P.R., Dalmijn W.L. Automatic sorting of scrap metals with a combined electromagnetic and dual energy X-ray transmission sensor // International Journal of Mineral Processing. 2007. Т. 82, вып. 4. С. 222-232. doi: 10.1016/j.minpro.2006.10.006
  43. Min Chen, Shiwen Mao, Yin Zhang, Victor C.M. Leung. Big Data. Related Technologies, Challenges, and Future Prospects. — Springer, 2014. — 100 p. — ISBN 978-3-319-06244-0. — doi:10.1007/978-3-319-06245-7
  44. MobileNet-Caffe. URL: <https://github.com/shicai/MobileNet-Caffe> (дата обращения: 15.03.2018).
  45. NIST Chemistry WebBook. URL: <https://webbook.nist.gov/chemistry/>(дата обращения: 15.03.2018).
  46. NIST, 2011, “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources

(e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”, p. 6.

47. P. Viola and M.J. Jones, Robust real-time face detection // International Journal of Computer Vision, Vol. 57, No. 2, 2004, P.137–154.
48. Pinal Dave. Big Data – Buzz Words: What is MapReduce – Day 7 of 21 [Электронный ресурс] // SQLauthority. URL: <https://blog.sqlauthority.com/2013/10/09/big-data-buzz-words-what-is-mapreduce-day-7-of-21/> (дата обращения 15.05.17)
49. Richard Zuech, Taghi M Khoshgoftaar, Randall Wald. Intrusion detection and Big Heterogeneous Data: a Survey // Journal of Big Data. 2015
50. Rosebrock A. Practical Python and OpenCV + Case Studies [Электронный ресурс]. – 2016. – Режим доступа: <https://www.pyimagesearch.com/2016/11/20/Python-OpenCV/> – Дата обращения 30 декабря 2017.
51. Rushil Anirudh, Jayaraman J. Thiagarajan, Bhavya Kailkhura & Peer-Timo Bremer. MimicGAN: Robust Projection onto Image Manifolds with Corruption Mimicking // International Journal of Computer Vision, 2020
52. Scherer, D. Evaluation of pooling operations in convolutional architectures for object recognition / D. Scherer, A. Müller, S. Behnke // Proceedings of International Conference on Artificial Neural Networks (ICANN). – 2010. – P. 92-101.
53. Scott D.M. A two-colour near-infrared sensor for sorting recycled plastic waste // Measurement Science and Technology. 1995. Т. 6, вып. 2. doi: 10.1088/0957-0233/6/2/004
54. Simchi-Levi, D.; Trick, M. A. Introduction to "Little's Law as Viewed on Its 50th Anniversary" (АНГЛ.) // Operations Research — 2013. — Vol. 59, no. 3. — P. 535. — doi:10.1287/opre.1110.0941

55. Simonyan K., Zisserman A.. Very deep convolutional networks for large-scale image recognition. arXiv препринт arXiv: 1409.1556, 2014. URL: <https://arxiv.org/pdf/1409.1556.pdf> (дата обращения: 16.12.2018).
56. Smith L. Cyclical Learning Rates for Training Neural Networks. arXiv препринт arXiv: 1506.01186, 2017. URL: <https://arxiv.org/pdf/1506.01186.pdf> (дата обращения: 20.09.2019).
57. Spectral Database for Organic Compounds SDBS. URL: [http://sdb.sdb.aist.go.jp/sdb/cgi-bin/cre\\_index.cgi](http://sdb.sdb.aist.go.jp/sdb/cgi-bin/cre_index.cgi)(дата обращения: 15.03.2018).
58. Supriya Suresh & Subaji Mohan. ROI-based feature learning for efficient true positive prediction using convolutional neural network for lung cancer diagnosis // Neural Computing and Applications, 2020
59. Supriya Suresh & Subaji Mohan. ROI-based feature learning for efficient true positive prediction using convolutional neural network for lung cancer diagnosis // Neural Computing and Applications, 2020
60. Synced. NVIDIA CEO Says “FGPA is Not the Right Answer” for Accelerating AI [Электронный ресурс]. URL: <https://medium.com/syncedreview/nvidia-ceo-says-fgpa-is-not-the-right-answer-for-accelerating-ai-83c810969edd>. (Дата обращения: 18.02.2018).
61. TensorFlow [Электронный ресурс]. URL: <https://www.tensorflow.org/> (дата обращения 04.05.2019).
62. Tobias Malach & Jitka Pomenkova. Optimal face templates: the next step in surveillance face recognition // Pattern Analysis and Applications, 2019
63. Transfer Learning. [Электронный ресурс] URL: <http://cs231n.github.io/transfer-learning/> (дата обращения: 15.03.2018).
64. Vasconcelos C.N., Vasconcelos B.N. Convolutional neural network committees for melanoma classification with classical and expert knowledge base image transforms data augmentation. arXiv препринт arXiv: 1702.07025, 2017. URL: <https://arxiv.org/pdf/1702.07025.pdf> (дата обращения: 16.12.2018).

65. Veita H.M., Diehla T.R., Salamia A.P., Rodriguesa J.S., Bernardesa A.M., Tenoriob J.A.S. Utilization of magnetic and electrostatic separation in the recycling of printed circuit boards scrap // Waste Management. 2005. Т. 25, вып. 1. С. 67-74. doi: 10.1016/j.wasman.2004.09.009657
66. Wang J., Perez L. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. URL: <http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf> (дата обращения: 16.12.2018).
67. Wang, Teng; Oral, Sarp; Wang, Yandong; Settlemyer, Brad; Atchley, Scott; Yu, Weikuan (October 2014). "BurstMem: A High-Performance Burst Buffer System for Scientific Applications". 2014 IEEE International Conference on Big Data (Big Data). IEEE. pp. 71–79.
68. Wani, M. Arif, Kantardzic, Mehmed, Sayed-Mouchaweh, Moamar (Eds.). Deep Learning Applications. 2020, ISBN 978-981-15-1816-4
69. Wani, M. Arif, Kantardzic, Mehmed, Sayed-Mouchaweh, Moamar (Eds.) // Advances in Deep Learning. 2020, ISBN 978-981-13-6794-6
70. X. Chan, Th. Gonnot, J. Saniie. Обнаружение и распознавание лиц в реальном времени в комплексе // Background Journal of Signal and Processing Information, 2017, 8, 99-112
71. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, Winter 1989.] [LeCun, Y. Deep learning / Y. LeCun, Y. Bengio, G. Hinton // Nature. – 2015. – Vol. 521. – No. 7553. – P. 436-444.
72. Yongbon Koo, Sunghoon Kim & Young-guk Ha. OpenCL-Darknet: implementation and optimization of OpenCL-based deep learning object detection framework // World Wide Web, 2020
73. Zak Stone Google's scalable supercomputers for machine learning, Cloud TPU Pods, are now publicly available in beta [Электронный ресурс]. URL:

- <https://cloud.google.com/blog/products/ai-machine-learning/googles-scalable-supercomputers-for-machine-learning-cloud-tpu-pods-are-now-publicly-available-in-beta>. (Дата обращения: 18.02.2018).
74. ZBar bar code reader. <http://zbar.sourceforge.net/>.
  75. Zeiler, M .D. Stochastic pooling for regularization of deep convolutional neural networks [Электронный ресурс] / M. D. Zeiler, R. Fergus // arXiv.org. – 2013. – Режим доступа: <http://arxiv.org/pdf/1301.3557.pdf>.
  76. Zemhari, Akka, Benois-Pineau, Jenny. Deep Learning in Mining of Visual Content. 2020, ISBN 978-3-030-34376-7
  77. Zhong Z., Zheng L., Kang G., Li S., Yang Y. Random Erasing Data Augmentation. arXiv препринт arXiv: 1708.04896, 2017. URL: <https://arxiv.org/pdf/1708.04896.pdf> (дата обращения: 16.12.2018).
  78. Алгоритмы обнаружения движения по видеопотоку. [Электрон. ресурс] URL: [http://translate.googleusercontent.com/translate\\_c?hl=ru&langpair=en%7Cru&rurl=translate.google.com&twu=1&u=http://www.codeproject.com/KB/audiovideo/Motion\\_Detection.aspx&usg=ALkJrhiuzz70ok1viVxDFDylMntEJEMT-A](http://translate.googleusercontent.com/translate_c?hl=ru&langpair=en%7Cru&rurl=translate.google.com&twu=1&u=http://www.codeproject.com/KB/audiovideo/Motion_Detection.aspx&usg=ALkJrhiuzz70ok1viVxDFDylMntEJEMT-A). (дата обращения: 15.03.2018)
  79. Ахметзянов К. Р., Южаков А. А. Оптимизация архитектуры нейронной сети для сортировки группы мусора // Известия СПбГЭТУ ЛЭТИ. - 2019. - № 7. - С. 62-70.
  80. Ахметзянов К. Р., Южаков А. А. Оптимизация архитектуры нейронной сети для сортировки мусорных отходов // Сборник статей по материалам Четвертой всероссийской научно-практической конференции, проводимой в рамках Пермского естественнонаучного форума «Математика и глобальные вызовы XXI века». - 2019. – Ч. 1. - С. 82-83.
  81. Ахметзянов К. Р., Южаков А. А. Разработка нейронной сети на основе знаний о воздействиях окружающей среды // Нейрокомпьютеры: разработка, применение. - 2019. – Т. 21. - № 3. - С. 5-13.

82. Ахметзянов К. Р., Южаков А. А. Сравнение сверточных нейронных сетей для задач сортировки мусорных отходов // Известия СПбГЭТУ ЛЭТИ. - 2018. - № 6. - С. 27-32.
83. Ахметзянов К. Р., Южаков А. А. Увеличение точности сверточной нейронной сети за счет возрастания количества данных // Нейрокомпьютеры: разработка, применение. - 2018. - № 7. - С. 14-19.
84. Бордунов В.В. и др. Перспективные полимерные волокнистые материалы для очистки воздуха // Известия Томского политехнического университета. – 2002. –Т. 305. –Вып. 3: Проблемы и перспективы технологий атомной промышленности: тематический выпуск. –С. 222–226.
85. Вентцель Е.С. Исследование операций. – М., «Советское радио», 1972, 552 стр.
86. Вентцель Е.С., Овчаров Л.А. Прикладные задачи теории вероятностей – М., «Радио и связь», 1983, 416 стр.
87. Визильтер Ю.В., Желтов С.Ю., Бондаренко А.В., Ососков М.В., Моржин А.В. Обработка и анализ изображений в задачах машинного зрения: Курс лекций и практических занятий. – М.: Физматкнига, 2010. – 672 с.
88. Виола Р., Джонс М.Дж. и Сноу Д.. Обнаружение пешеходов с использованием моделей движения и внешнего вида. 9-й ICCV, Ницца, Франция, том 1, страницы 734–741, 2003.
89. Вторичные материальные ресурсы [Электронный ресурс] URL: <http://www.ekoslovar.ru/056.html> (дата обращения 03.01.2017).
90. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – Техносфера, 2005. – 1072 с. Петровский И. Г. Лекции по теории обыкновенных дифференциальных уравнений. – Изд. 7-е, испр. и доп. Изд-во МГУ, 1984. – 295 с.
91. Джулли А., Пал С. Библиотека Keras — инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow = Deep learning with Keras. — ДМК-Пресс, 2017. — 294 с. — ISBN 978-5-97060-573-8.

92. Дятлов Е. И. Машинное зрение (аналитический обзор) // Математические машины и системы, 2013 [режим доступа: <https://cyberleninka.ru/article/n/mashinnoe-zrenie-analiticheskiy-obzor>]
93. Зелёные IT - экология в мире информационных технологий [Электронный ресурс]. URL: <http://www.dal.by/news/149/04-10-14-8/> (Дата обращения: 18.02.2018).
94. Иванько М. А. Клепикова А.В. Системы искусственного зрения // Вестник МГУП имени Ивана Федорова – 2015. – №5.
95. Клинков А.С. и др. Утилизация и вторичная переработка тары и упаковки из полимерных материалов. –Тамбов: Изд. ТГТУ, 2010. –100 с.
96. Кокоулин А. Методы распределенной обработки и хранения больших изображений (2013) IEEE EuroCon 2013, статья № 6625191, с. 1606-1610. DOI: 10.1109 / EUROCON.2013.6625191
97. Кокоулин А. Н., Тур А. И., Даденков С. А. Оптическая система сортировки пустых контейнеров // Нейрокомпьютеры: разработка, применение – 2018. – №7. – С. 3-7.
98. Кокоулин А. Н., Тур А. И., Князев А. И., Южаков А. А. Вопросы разработки и настройки оптической подсистемы распознавания и сортировки пустых контейнеров в системе раздельного приема бытовых отходов // Информационно-измерительные и управляющие системы применение – 2018. – №9. – Т. 16. – С. 45-50 [DOI: 10.18127/j20700814-201809-07].
99. Кокоулин А.Н., Южаков А.А., Кирьянов Д.А. Масштабируемое распределенное хранилище для больших научных данных (2018 г.) Материалы конференции молодых ученых России в области электротехники и электроники 2018 г., ElConRus 2018, январь 2018 г., с. 1099 -1103. DOI: 10.1109 / ElConRus.2018.8317282
100. Кокоулин, А.Н. Подход распределенного хранения в сетях доставки контента (2017) 2016 13-я Международная научно-техническая конференция по актуальным проблемам электронного приборостроения,

АРЕИЕ 2016 - Proceedings, 2, статья № 7806398, с. 479-484. DOI: 10.1109 / АРЕИЕ.2016.7806398

101. Кокоулин А. Методы распределенной обработки и хранения больших изображений (2013) IEEE EuroCon 2013, статья № 6625191, с. 1606-1610. DOI: 10.1109 / EUROCON.2013.6625191
102. Косинцев В.И. и др. Антимикробные волокнистые материалы // Энергетика: экология, надежность, безопасность: Материалы докладов XV Всероссийской научно-техн. конф. –Томск, 9–11 декабря 2009. –Томск: Изд-во ТПУ, 2009.
103. Косинцев В.И. и др. Волокнистые материалы для очистки воды от нефтепродуктов // Энергетика: экология, надежность, безопасность: Материалы докладов XV Всероссийской научно-техн. конф. –Томск, 9–11 декабря 2009. –Томск: Издательство ТПУ, 2009.
104. Куликов А.А., Демкин Д.В., Мельков А.Е. Анализ влияния максимального сжатия лица на результат распознавания лица. ПЕРСПЕКТИВЫ НАУКИ (ISSN 2077-6810). № 3 (54), Тамбов, Россия, 2014, с. 104-108
105. Майер-Шенбергер Виктор, Кеннет Кукьер. Большие данные. Революция, которая изменит то, как мы живём, работаем и мыслим = Big Data. A Revolution That Will Transform How We Live, Work, and Think / пер. с англ. Инны Гайдюк. — М.: Манн, Иванов, Фербер, 2014. — 240 с. — ISBN 987-5-91657-936-9.
106. Масленников А. Вторая жизнь // PakkoGraff.– 2004. –№ 8. URL: <http://www.pakkograff.ru/reader/articles/materials/polymers/1069.php> (дата обращения: 10.03.2012).
107. Методы компьютерной обработки изображений под редакцией В.А. Сойфера. М.: Физматлит, 2003. 784 с.
108. Мусорные богатства России. [Электронный ресурс]. URL: <https://www.rbc.ru/newspaper/2013/03/22/56c1b6129a7947406ea09ec6/> (дата обращения: 15.03.2018).

109. Огрель Л. Д. Оценка накопления, сбора и переработки отходов ПЭТФ в России // Экологический вестник России – 2012. – №4. – С. 26-31.
110. Одесс В.И. Вторичные ресурсы: хозяйственный механизм использования. –М.: Экономика, 1988. –160 с.
111. Орельен Жерон. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. Концепции, инструменты и техники для создания интеллектуальных систем = Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques for Building Intelligent Systems. — Вильямс, 2018. — 688 с. — ISBN 978-5-9500296-2-2, 978-1-491-96229-9.
112. Орлов К.Е., Нефедов В.Д., Попов В.В. Система отслеживания брака на производстве на основе машинного зрения // Сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых учёных. Под редакцией Т. Е. Мамоновой. 2016, С. 215-216 [доступ: <https://www.elibrary.ru/item.asp?id=25796182>]
113. Переработка отходов ПЭТ // GalPET official site. 2007. URL: <http://www.galpet.com.ua/pererabotka-othodov-pet#a4> (дата обращения: 12.03.2012).
114. Подсчитано общее количество произведенного человеком пластика. URL: <https://chr.dk.ru/news/podschitano-obshchee-kolichestvo-plastika> (дата обращения: 15.03.2018).
115. Пономарёва В.Т., Лихачёва Н.Н., Ткачик З.А. Использование пластмассовых отходов за рубежом // Пластические массы. – 2002. –№ 5. – С. 44–48.
116. Сконников П., Покровский П. Процедура выделения контуров объекта на изображении для распознавания в системах технического зрения мобильных роботизированных платформ // СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ - СТНО-2018 Сборник трудов международного научно-технического форума: в 11 томах. Под

- общ. ред. О.В. Миловзорова, 2018, С. 209-214.  
[<https://www.elibrary.ru/item.asp?id=35399075>]
117. Сконников П., Покровский П. Процедура выделения контуров объекта на изображении для распознавания в системах технического зрения мобильных роботизированных платформ // СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ - СТНО-2018 Сборник трудов международного научно-технического форума: в 11 томах. Под общ. ред. О.В. Миловзорова, 2018, С. 209-214.  
[<https://www.elibrary.ru/item.asp?id=35399075>]
118. Сроки разложения отходов: сколько поколений переживет Ваша мусорная корзина?. URL: <http://eco-boom.com/sroki-razlozheniya-othodov-skolko-pokolenij-perezhivet-vasha-musornaya-korzina/> (дата обращения: 15.03.2018).
119. Стельмах С. Edge computing: почему об этой технологии следует узнать немедленно? [Электронный ресурс]. URL: <https://www.itweek.ru/iot/article/detail.php?ID=198653> (Дата обращения: 18.02.2018).
120. Танг Дж., Денг С., Хуан Г.Б. и Чжао Б. (2015) Обнаружение судов в сжатом домене на космическом оптическом изображении с использованием глубокой нейронной сети и машины экстремального обучения. IEEE Труды по геонаукам и дистанционному зондированию, 53, 1174-1185. [<https://doi.org/10.1109/TGRS.2014.233575>].
121. Труфанова Е.В., Кузнецова Н.А., Макеев П.В. Штрихкод. История создания, его разновидности и назначение // Новая наука: теоретический и практический взгляд. 2016. №8. С. 186-188.
122. Тур А.И. Использование технологии burst buffer для обработки больших данных (Перспективные технологии в средствах передачи информации. Т. 1)
123. Тур А.И. Применение каскадов Хаара для распознавания объекта (Автоматизированные системы управления и информационные технологии. Т. 1

124. Тютляева Е. О., Тютляев М. М. Системы, хранения данных лидирующих суперкомпьютеров // Программные системы: теория и приложения: электронный научный журнал. 2014
125. Чезари Л. Асимптотическое поведение и устойчивость решений обыкновенных дифференциальных уравнений. – М.: Мир, 1964. – 477 с.
126. Черняк Л. Интеграция – основа облака [Электронный ресурс]. URL: <http://www.osp.ru/os/2011/07/13010473/> (Дата обращения: 18.02.2018).
127. Южаков А. А., Кокоулин А. Н., Тур А. И. Иерархическая архитектура сверточной нейронной сети в распределенной системе распознавания лиц // Нейрокомпьютеры: разработка, применение.- 2019. – Т. 21, № 3. – С. 28-34. - DOI 10.18127/j19998554-201903-04.
128. Южаков А.А., Тур А.И. Выбор алгоритмов для реализации системы машинного зрения // Материалы XIV Всероссийской школы-конференции молодых ученых – 2017. – С. 377-384.
129. Яне Б. Цифровая обработка изображений – М.: Техносфера, 2007. – 584с.

## Приложение А

Таблица знаков и кодов переработки

Знак	Пояснение
	<p><b>Нетоксичный материал</b> — изделие изготовлено из материала (обычно — пластик), который нетоксичен и может соприкасаться с пищевыми продуктами.</p>
	<p><b>Знак «Вредно для здоровья»</b>. Обычно помещается на упаковках бытовой химии, в частности, средствах для посудомоечных машин.</p>
	<p><b>«Выбросить в урну»</b> — именно это и следует сделать с упаковкой, на которой есть такая экомаркировка.</p>
	<p><b>Особая утилизация</b>. Ставится на источниках питания (батарейки) и товарах, содержащих некоторые опасные вещества (ртуть, свинец). Во избежание нанесения вреда окружающей среде необходимо отделить данный объект от обычных отходов и утилизировать его наиболее безопасным способом — например, сдать в специальные места по утилизации.</p>
	<p><b>Знак вторичной переработки «Петля Мебиуса»</b> означает, что упаковка товара частично или полностью сделана из переработанного сырья либо пригодна для последующей переработки.</p>
	<p><b>Знак «Зеленая точка» (нем. <i>Der Grüne Punkt</i>)</b> ставят на продукцию, производитель которой оплатил сбор на переработку и утилизацию в рамках «Дуальной системы» (DSD).</p>

## Продолжение таблицы знаков и кодов переработки

Знак	Пояснение
	<p>Знак соответствия <b>Системы экологической сертификации (РОСС RU.001.01.ЭТОО).</b></p>
	<p><b>Потенциально перерабатываемый пластик</b> — знак ставится непосредственно на изделия.</p> <ol style="list-style-type: none"> <li>1. <b>PET</b> или <b>PETE</b> (<b>ПЭТ</b>) — полиэтилентерфталат;</li> <li>2. <b>PEHD</b> или <b>HDPE</b> (<b>ПЭНД</b>) — полиэтилен высокой плотности или низкого давления</li> <li>3. <b>PVC</b> (<b>ПВХ</b>) — поливинилхлорид</li> <li>4. <b>LDPE</b> или <b>PELD</b> (<b>ПЭВД</b>) — полиэтилен низкой плотности, полиэтилен высокого давления</li> <li>5. <b>PP</b> (<b>ПП</b>) — полипропилен</li> <li>6. <b>PS</b> (<b>ПС</b>) — полистирол</li> <li>7. <b>OTHER</b> или <b>О</b> — другие виды пластика (в основном, поликарбонат — может содержать опасный для человека бисфенол)</li> </ol>
	<p><b>Изготовлено из алюминия</b>, который подлежит вторичной переработке.</p>
	<p><b>Знак переработки целлюлозной продукции:</b></p> <ol style="list-style-type: none"> <li>20 — картон;</li> <li>21 — прочая бумага (журналы, почта, упаковка из-под муки, сахара и т.д.)</li> <li>22 — бумага</li> <li>23 — полиграфический картон (открытки, обложки книг и пр.)</li> </ol>

## Продолжение таблицы знаков и кодов переработки

Знак	Пояснение
	<p><b>Знак переработки стекла:</b></p> <p><b>70</b> — бесцветное стекло</p> <p><b>71</b> — зеленое стекло</p> <p><b>72</b> — коричневое стекло</p> <p><b>73</b> — бутылочное стекло (тёмно-коричневое, Dark Sort Glass)</p> <p><b>74</b> — бутылочное стекло (светло-коричневое, Light Sort Glass)</p> <p><b>75</b> — стекло с малым содержанием свинца (используется в современных телевизорах и электронных приборах)</p> <p><b>76</b> — хрусталь</p> <p><b>77</b> — стекло, покрытое медью</p> <p><b>78</b> — стекло, покрытое серебром</p> <p><b>79</b> — позолоченное стекло</p>
	<p><b>Композитные материалы</b></p> <p><b>81</b> — бумага + пластик (пакеты для кормов животных, одноразовые тарелки и т.п.)</p> <p><b>82</b> — бумага / фибровый картон + алюминий</p> <p><b>83</b> — бумага / фибровый картон + жесь</p> <p><b>84</b> — бумага / картон + пластик + алюминий (контейнеры для жидкостей, упаковки для сока, обёртки жвачек, Tetra Brik)</p> <p><b>85</b> — бумага / фибровый картон + пластик + алюминий + жесь</p> <p><b>87</b> — биоразлагаемый пластик</p>

Окончание приложения А

## Окончание таблицы знаков и кодов переработки

<b>Знак</b>	<b>Пояснение</b>
	<p><b>90</b> — пластик + алюминий</p> <p><b>91</b> — пластик + жечь</p> <p><b>92</b> — пластик + разные металлы</p> <p><b>95</b> — стекло + пластик</p> <p><b>96</b> — стекло + алюминий</p> <p><b>97</b> — стекло + жечь</p> <p><b>98</b> — пластик + разные металлы.</p>

**Приложение Б**

Пример листинга расчёта основных показателей одноканальной СМО,

$$\text{alpha} := 11$$

$$\text{mu} := \frac{60000000}{2073600}$$

$$\text{ro} := \frac{\text{alpha}}{\text{mu}}$$

$$p0 := \frac{\text{mu}}{\text{alpha} + \text{mu}}$$

$$p1 := \frac{\text{alpha}}{\text{alpha} + \text{mu}}$$

$$q := \frac{\text{mu}}{\text{alpha} + \text{mu}}$$

$$\text{otk} := \frac{\text{alpha}}{\text{alpha} + \text{mu}}$$

$$a := \frac{\text{alpha} \cdot \text{mu}}{\text{alpha} + \text{mu}}$$

Относительная пропускная способность

$$q = 0.725$$

Вероятность отказа

$$\text{otk} = 0.275$$

Абсолютная пропускная способность

$$a = 7.97$$



**Приложение Г**

Код программы для ЭВМ

**«ПРОГРАММА БИНОКУЛЯРНОГО ЗРЕНИЯ С УЧЕТОМ РАССТОЯНИЯ  
ДО ОБЪЕКТА»**

```
int trigPin = 11;
int echoPin = 12;
long duration, mm, cm, inches;
String str = "";

void setup() {
    Serial.begin (9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop() {
    str = Serial.readString();

    if (str == "ok\n"){
        digitalWrite(trigPin, LOW);
        delayMicroseconds(5);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
```

```
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);

mm = (duration/2) / 2.91;

Serial.print(mm);
Serial.print("mm");
Serial.println();
}
}

import traceback
import argparse
import cv2
import sys

from distance import Distance
from image import Image
from panorama_stitching import Stitcher
from dictionary import Dictionary

args = None

def createMatrix():
    global args
    dist = Distance()
```

```
image = Image(args.camera1, args.camera2)
stitcher = Stitcher()
dictionary = Dictionary()
while True:
    d = dist.getDistance()
    image.getImages()
    if args.debug:
        print("Distance: ", d)
    key = cv2.waitKey(1)
    if key & 0xFF == 13:
        res, H = stitcher.stitch(image.getImages())
        if res is not None:
            cv2.imshow("Result", res)
            dictionary[d] = H
            dictionary.save()
    elif key & 0xFF == ord('q'):
        break

def loadMatrix():
    global args
    dist = Distance()
    image = Image(args.camera1, args.camera2)
    stitcher = Stitcher()
    dictionary = Dictionary()
    dictionary.load()
```

```
while True:
    d = dist.getDistance()
    image.getImages()
    if args.debug:
        print("Distance: ", d)
    key = cv2.waitKey(1)
    if key & 0xFF == 13:
        minV = 9999999.0
        closestD = 0
        for key in dictionary.keys():
            if abs(key - d) < minV:
                minV = abs(key - d)
                closestD = key
        if args.debug:
            print("Closest distance: ", closestD)
        H = dictionary[closestD]
        res = stitcher.stitch(image.getImages(), isTuned =
False, H = H)
        if res is not None:
            cv2.imshow("Result", res)
    elif key & 0xFF == ord('q'):
        break

if __name__ == "__main__":
    try:
```

```
parser = argparse.ArgumentParser()
parser.add_argument(
    "--camera1",
    default = "2")
parser.add_argument(
    "--camera2",
    default = "1")
parser.add_argument(
    "--create",
    action="store_true",
    default=False
)
parser.add_argument(
    "--debug",
    action="store_true",
    default=False
)
args = parser.parse_args()
if args.create:
    createMatrix()
else:
    loadMatrix()
except:
    print ("Unexpected error:", sys.exc_info()[0])
    traceback.print_exc(file=sys.stdout)
```

```
import pickle

class Dictionary:
    def __init__(self):
        self.d = {}

    def __setitem__(self, key, value):
        self.d[key] = value

    def __getitem__(self, key):
        return self.d[key]

    def keys(self):
        return self.d.keys()

    def save(self):
        output = open('H.pkl', 'wb')
        pickle.dump(self.d, output)
        output.close()

    def load(self):
        pkl_file = open('H.pkl', 'rb')
        self.d = pickle.load(pkl_file)
        pkl_file.close()

import serial

import time

class Distance:
    def __init__(self):
        self.ser = serial.Serial('/dev/ttyACM0', 9600)
```

```
        time.sleep(2)
def __del__(self):
    self.ser.close()
def getDistance(self):
    if(self.ser.isOpen()):
        self.ser.write("ok\n".encode("utf-8"))
        incData = self.ser.readline()
        d = incData.decode("utf-8")[:-1].rsplit()[0][:-2]
        return int(d)
    else:
        raise Exception("Port not open")

import cv2
class Image:
    def __init__(self, cameraId1, cameraId2):
        self.cap1 = cv2.VideoCapture(int(cameraId1))
        self.cap2 = cv2.VideoCapture(int(cameraId2))
        if not(self.cap1.isOpened()):
            raise Exception("Camera #1 not open")
        if not(self.cap2.isOpened()):
            raise Exception("Camera #2 not open")
    def __del__(self):
        self.cap1.release()
        self.cap2.release()
        cv2.destroyAllWindows()
```

```
def getImages(self):
    img1, img2 = None, None
    if self.cap1.isOpened():
        ret1, img1 = self.cap1.read()
        if not(ret1):
            raise Exception("Image from Camera #1 not read")
        cv2.imshow('Frame1', img1)
    else:
        raise Exception("Camera #1 not open")
    if self.cap2.isOpened():
        ret2, img2 = self.cap2.read()
        if not(ret2):
            raise Exception("Image from Camera #2 not read")
        cv2.imshow('Frame2', img2)
    else:
        raise Exception("Camera #2 not open")
    return img1, img2
```

```
import numpy as np
import cv2
```

```
class Stitcher:
    imageA = None
    imageB = None
    isTuned = None
```

```
ratio = None

reprojThresh = None

H = None

def stitch(self, images, isTuned = True, ratio = 0.75,
reprojThresh = 4.0, H = None):

    (self.imageB, self.imageA) = images

    self.isTuned = isTuned

    self.ratio = ratio

    self.reprojThresh = reprojThresh

    self.H = H

    if isTuned:

        return self.tune()

    else:

        return self.concatImages(self.H)

def tune(self):

    (kpsA, featuresA) = self.detectAndDescribe(self.imageA)

    (kpsB, featuresB) = self.detectAndDescribe(self.imageB)

    M = self.matchKeypoints(kpsA, kpsB, featuresA,
featuresB, self.ratio, self.reprojThresh)

    if M is None:

        return None

    (matches, H, status) = M
```

```
result = self.concatImages(H)

return result, H

def concatImages(self, H):

    result = cv2.warpPerspective(self.imageA, H,
    (self.imageA.shape[1] + self.imageB.shape[1],
    self.imageA.shape[0]))

    result[0:self.imageB.shape[0], 0:self.imageB.shape[1]]
= self.imageB

    return result

def detectAndDescribe(self, image):

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    descriptor = cv2.xfeatures2d.SIFT_create()

    (kps, features) = descriptor.detectAndCompute(image,
None)

    kps = np.float32([kp.pt for kp in kps])

    return (kps, features)

def matchKeypoints(self, kpsA, kpsB, featuresA,
featuresB, ratio, reprojThresh):

    matcher = cv2.DescriptorMatcher_create("BruteForce")

    rawMatches = matcher.knnMatch(featuresA, featuresB, 2)

    matches = []
```

```
for m in rawMatches:
    if len(m) == 2 and m[0].distance < m[1].distance *
ratio:
        matches.append((m[0].trainIdx, m[0].queryIdx))
if len(matches) > 4:
    ptsA = np.float32([kpsA[i] for (_, i) in matches])
    ptsB = np.float32([kpsB[i] for (i, _) in matches])

    (H, status) = cv2.findHomography(ptsA, ptsB,
cv2.RANSAC, reprojThresh)
    return (matches, H, status)

return None
```

## Приложение Д

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.

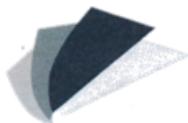
Руководитель Федеральной службы  
по интеллектуальной собственности

*Г.П. Ивлиев* Г.П. Ивлиев



## Приложение Е

Акт внедрения результатов кандидатской диссертационной работы в производство



Общество с ограниченной ответственностью  
«ГОРНЫЕ ВЕНТИЛЯЦИОННЫЕ УСТРОЙСТВА»

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

06.10.2020 № 2020/10-1

### АКТ

о внедрении результатов кандидатской диссертационной работы  
Тур Александра Игоревича

Комиссия в составе:

Председатель комиссии: Николаев А.В., генеральный директор ООО «Горные вентиляционные устройства», канд. тех. наук.

Член комиссии: Польшгалов С.В., специалист отдела проектирования и разработки, канд. тех. наук.

составила настоящий акт о том, что результаты диссертационной работы «ИЕРАРХИЧЕСКИЙ МЕТОД РАСПОЗНАВАНИЯ В ПОДСИСТЕМАХ МАШИННОГО ЗРЕНИЯ АСУТП СОРТИРОВКИ И УТИЛИЗАЦИИ БЫТОВЫХ ОТХОДОВ» использованы при проектировании и реализации системы оптического распознавания в АПК «Sortomat».

При создании модификации АПК «Sortomat 1.0» были использованы и внедрены результаты диссертационной работы – изменён процесс распознавания тары в АПК «Sortomat» в соответствии с предложенным автором методом иерархического распознавания:

- разработанные модели (аналитическая и имитационная) были применены для расчёта характеристик системы, подходящих для иерархического распознавания;
- реализован алгоритм, выполняющий поэтапное уточнение местоположения тары на изображении, получаемого от камеры отсека распознавания, с использованием иерархического метода.

Применение результатов диссертационной работы обеспечивает уменьшение времени распознавания тары в среднем на 70% по сравнению с предшествующими показателями, уменьшилось количество случаев некорректной работы микроконтроллера Raspberry Pi3, вызванных зависанием алгоритма распознавания при анализе изображения.

Перераспределение освободившихся вычислительных мощностей позволило внести конструктивные изменения в отсек распознавания тары:

- уменьшена высота отсека на 50%;
- основная камера смещена вдоль оси отсека распознавания;

- установлена дополнительная камера.

Уменьшение высоты отсека распознавания позволило увеличить полезный объем отсека хранения, что положительно сказывается на времени автономной работы АПК «Sortomat» и уменьшает время его самокупаемости.

Кроме того, предложенные Тур А.И. в его диссертационной работе решения будут использованы при проектировании новой модели АПК «Sortomat».

Председатель комиссии



А.В. Николаев

Член комиссии

С.В. Полюгалов

## Приложение Ж

Акт внедрения результатов кандидатской диссертационной работы в уч. процесс

УТВЕРЖДАЮ

Проректор по учебной работе  
Пермского национального  
исследовательского политехнического  
университета



/ Н.В. Лобов /

13 » сентября 2020 г.

### АКТ

о внедрении результатов,

полученных Туром Александром Игоревичем

при выполнении диссертационной работы

на соискание ученой степени кандидата технических наук

«Иерархический метод распознавания в подсистемах машинного зрения

АСУТП сортировки и утилизации 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»,  
Кокоулин Андрей Николаевич, кандидат технических наук, доцент кафедры «Автоматика и телемеханика»,

составила настоящий акт о том, что результаты диссертационной работы «Иерархический метод распознавания в подсистемах машинного зрения АСУТП сортировки и утилизации бытовых отходов» соискателя Тура А.И. 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 построения и функционирования систем машинного зрения с учетом ограниченных вычислительных возможностей вычислительной платформы нашли применение:

- 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 поэтапного уточнения информации на изображении, позволяющей повысить быстродействие системы без снижения достоверности распознавания и вычислительной сложности алгоритма.

- 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 нейронных сетей с большой вычислительной сложностью 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 и реализации 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



/ Кокоулин А. Н. /

« 05 » октября 2020 г.