

## СОДЕРЖАНИЕ

<b>СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ .....</b>	<b>10</b>
<b>ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ .....</b>	<b>12</b>
<b>ВВЕДЕНИЕ.....</b>	<b>14</b>
<b>1 ОБЗОР ТЕКУЩИХ ПРОИЗВОДСТВЕННЫХ РЕШЕНИЙ И СПОСОБЫ ИХ ОПТИМИЗАЦИИ. АНАЛИЗ АЛЬТЕРНАТИВНЫХ СПОСОБОВ ОПРЕДЕЛЕНИЯ ОБЪЕМНО ВЕСОВЫХ ХАРАКТЕРИСТИК .....</b>	<b>17</b>
1.1 Анализ предметной области .....	18
1.1.1 Устройство современных складов .....	18
1.1.2 Движение посылки внутри склада.....	19
1.1.3 Текущие технологические решения складской логистики .....	21
1.2 Формулирование проблем связанных с определением весогабаритных характеристик .....	23
1.3 Технические способы определения весогабаритных характеристик.....	26
1.3.1 Датчики расстояния.....	26
1.3.2 Техническое зрение .....	29
1.3.3 Лидар.....	32
1.4 Выбор технологии для дальнейшего исследования.....	35
1.5 Вывод к главе .....	37
<b>2 ИСПОЛЬЗОВАНИЕ ГЛУБОКОГО ОБУЧЕНИЯ И НЕЙРОННЫХ СЕТЕЙ ДЛЯ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ И ДЕТЕКЦИИ ТОВАРОВ НА СОРТИРОВОЧНОЙ ЛЕНТЕ. ....</b>	<b>38</b>
2.1 Анализ применения глубокого обучения и нейронных сетей в области складской логистики.....	39
2.1.1 Формулирование сценариев применения глубокого обучения и нейронных сетей в области складской логистики .....	39
2.1.2 Сведение сценариев применения глубокого обучения и нейронных сетей к классическим задачам компьютерного зрения.....	40

2.2	Применение глубокого обучения для решения задач складской логистики .....	41
2.2.1	Модель компьютерного зрения Yolo.....	41
2.2.2	Обучение модели Yolo на пользовательском наборе данных OpenImages.....	42
2.2.3	Использование модели MediaPipe для детекция 3d-объектов .....	51
2.3	Вывод к главе .....	53
<b>3</b>	<b>РАЗРАБОТКА СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ ДЕТЕКЦИИ И СЕГМЕНТАЦИИ ТОВАРОВ НА СОРТИРОВОЧНОЙ ЛЕНТЕ .....</b>	<b>55</b>
3.1	Методы повышения точности детектирования объектов на изображениях.....	56
3.1.1	Улучшение предварительной обработки изображений.....	56
3.1.2	Оптимизация гиперпараметров.....	57
3.1.3	Применение синтетических и дополненных данных.....	58
3.2	Архитектурный анализ и проектирование системы .....	60
3.2.1	Диаграмма IDEF0 .....	60
3.2.2	Диаграмма развертывания .....	68
3.2.3	Диаграмма классов .....	69
3.2.4	Архитектура возможной интеграции в существующие системы.....	71
3.3	Разработка модуля компьютерного зрения для решения задач складской логистики .....	72
3.3.1	Решение задачи детекции .....	72
3.3.2	Решение задачи сегментации .....	79
3.3.3	Создание синтетической тестовой среды в редакторе трехмерной графики Blender .....	84
3.3.4	Алгоритм определения облака точек объекта .....	87
3.3.4.1	Детальная сегментация Segmentation Anything .....	89
3.3.4.2	Детальная сегментация OpenCV .....	91

3.3.5	Преобразование облака точек в детализированные элементы объекта	93
3.3.6	Расчет габаритных характеристик товара на ленте .....	96
3.4	Вывод к главе .....	98
<b>ЗАКЛЮЧЕНИЕ .....</b>		<b>101</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>		<b>102</b>

## СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

**VBOX** – Bounding box. Прямоугольная рамка, используемая для обозначения положения и размера объекта на изображении. В задачах детекции объектов bounding box обводит обнаруженные объекты.

**CNN** – Convolutional Neural Network. Сверточная нейронная сеть, тип нейронной сети, особенно эффективный для задач компьютерного зрения, таких как классификация изображений и детекция объектов.

**COCO** – Common Objects in Context. Крупный датасет для задач компьютерного зрения, включающий изображения с аннотациями для детекции, сегментации и описания объектов. COCO используется для обучения и тестирования моделей машинного обучения.

**GAN** – Generative Adversarial Network. Архитектура нейронных сетей, состоящая из двух моделей: генератора и дискриминатора. Эти две модели соревнуются друг с другом в процессе обучения.

**IoU** – Intersection over Union. Метрика, используемая для оценки точности детекции объектов. Она вычисляется как площадь пересечения предсказанной и истинной bounding box, деленная на площадь их объединения.

**LIDAR** – Light identification, detection and ranging. Технология дистанционного зондирования, использующая лазерные лучи для измерения расстояний до объектов и создания высокоточных трехмерных моделей окружающего пространства.

**NMS** – Non-Maximum Suppression. Метод, используемый для удаления избыточных bounding box в процессе детекции объектов. Он оставляет только те bounding box, которые соответствуют наиболее вероятным местоположениям объектов.

**R-CNN** – Region-based Convolutional Neural Network. Семейство методов для детекции объектов, которое включает такие методы, как Fast R-CNN и Faster R-CNN. Эти методы используют региональные предложения для локализации объектов перед их классификацией.

SAM – Segment Anything Model. Модель компьютерного зрения от Meta AI, обученная на датасете из 11 млн картинок и 1.1 млрд масок сегментации. Предназначена для выполнения задачи сегментации любых объектов на изображении.

YOLO – You Only Look Once. Архитектура сверточной нейронной сети для детекции объектов в реальном времени. YOLO обрабатывает все изображение сразу, что делает ее быстрой и эффективной.

OBX – Объемно–весовые характеристики. Характеристики, описывающая габаритные и весовые свойства объекта. Используется в логистике для оптимизации упаковки и транспортировки.

## ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Аннотация – процесс маркировки или разметки входных данных при подготовке к обучению ИИ. В компьютерном зрении входные изображения и видео должны быть аннотированы в соответствии с задачей, которую должна выполнять модель.

Гиперпараметры – параметры модели машинного обучения, значения которых задаются до начала обучения и не изменяются в процессе обучения. Примеры гиперпараметров включают скорость обучения, количество слоев нейронной сети и размер пакета.

Датасет — это совокупность данных, собранных и организованных для конкретной цели, часто используемых в машинном обучении и искусственном интеллекте для обучения, тестирования и оценки моделей.

Детекция – процесс обнаружения объектов или особенностей на изображении или в видеопотоке. В контексте компьютерного зрения это обычно означает определение наличия и местоположения объектов.

Компьютерное зрение – область искусственного интеллекта, которая позволяет компьютерам интерпретировать и понимать визуальную информацию из окружающего мира, используя цифровые изображения и видео.

Маска – в контексте компьютерного зрения, маска представляет собой изображение того же размера, что и исходное изображение, где каждый пиксель соответствует конкретной категории или объекту. Маска используется в задачах сегментации, чтобы выделить пиксели, принадлежащие определенному объекту.

Сегментация – процесс разделения изображения на несколько сегментов (областей) для упрощения анализа. В компьютерном зрении сегментация используется для выделения объектов или областей интереса на изображении.

Синтетическая среда – виртуальная среда, созданная с использованием компьютерных технологий, которая имитирует реальные условия. Используется для тестирования и обучения систем искусственного интеллекта в контролируемых условиях.

Синтетические данные – данные, созданные искусственно, а не собранные из реальных источников. Они используются для обучения моделей машинного обучения, особенно когда реальные данные недоступны или их недостаточно.

Сортер – устройство или система, предназначенные для автоматического разделения или сортировки предметов по определенным характеристикам, таким как размер, вес, цвет или форма.

Техническое зрение – термин, используемый для описания применения компьютерного зрения в промышленности для автоматизации процессов контроля качества, управления роботами и других задач.

Фулфилмент – процесс выполнения заказа, включающий хранение товаров, обработку заказов, упаковку и доставку конечному потребителю.

## ВВЕДЕНИЕ

Современные технологии и методы оптимизации производственных процессов играют ключевую роль в современной промышленности. В контексте складской логистики особое внимание уделяется оптимизации сложных процессов, что имеет прямое влияние на эффективность складских операций. Современная складская логистика сталкивается с постоянными вызовами, требующими поиска новых технологических решений для оптимизации процессов. Введение систем компьютерного зрения в складскую логистику обладает значительным потенциалом для улучшения эффективности операций и повышения точности управления.

На сегодняшний день большинство доступных систем компьютерного зрения недостаточно адаптированы для выполнения задач по автоматизации складской логистики. В свою очередь специализированные системы являются недоступными для широкой публики. Отсутствие доступных open-source проектов ограничивает возможности разработки новых решений. Коммерческие предложения, в свою очередь, стоят дорого и недоступны для многих предприятий, что затрудняет внедрение современных технологий в сферу складской логистики. Это создаёт необходимость в разработке более доступных, специализированных систем.

Основной проблемой, решаемой в данной работе, является разработка системы компьютерного зрения, способной автоматизировать процесс детекции и сегментации товаров на сортировочной ленте, что позволит повысить эффективность и точность логистических операций, а также снизить их стоимость.



Целью данной работы является создание и разработка системы компьютерного зрения для детекции и сегментации товаров на сортировочной ленте. Для достижения этой цели необходимо решить следующие задачи:

- Анализ существующих решений и определение их недостатков.
- Разработка алгоритмов детекции и сегментации товаров с использованием методов машинного обучения и глубоких нейронных сетей.
- Реализация программного обеспечения для обработки и анализа изображений товаров.

Методы исследования включают в себя анализ существующих технологий, разработку и адаптацию алгоритмов глубокого обучения, проведение экспериментов с использованием синтетических данных, а также дообучение моделей на пользовательских датасетах.

Методологической основой исследования являются современные подходы к разработке систем компьютерного зрения и методы глубокого обучения. Теоретическая основа включает исследования в области компьютерного зрения, машинного обучения и нейронных сетей, а также принципы оптимизации технологических процессов в складской логистике.

Разработанная система позволит предприятиям сократить издержки, связанные с логистическими процессами, уменьшить число ошибок при сортировке товаров и повысить общую производительность. Это, в свою очередь, способствует повышению конкурентоспособности и ускорению внедрения новых технологий в складскую логистику.

Первая глава посвящена анализу предметной области, включая устройство современных складов, движение посылки внутри склада и текущие технологические решения складской логистики. Формулируются проблемы, связанные с определением габаритных характеристик, и рассматриваются технические способы их определения, включая использование датчиков расстояния, технического зрения и ЛИДАР.

Вторая глава сосредотачивается на использовании современных методов машинного обучения, в частности глубокого обучения и нейронных сетей, для решения задач складской логистики, таких как автоматическое распознавание и детекция товаров на сортировочной ленте. Производится анализ применения этих методов в области складской логистики, формулируются сценарии их применения, а также сводятся сценарии применения к классическим задачам компьютерного зрения. В рамках главы рассматривается применение модели YOLO для детекции товаров, обучение этой модели на пользовательском наборе данных OpenImages, а также использование модели MediaPipe для детекции 3D-объектов.

Третья глава посвящена разработке системы компьютерного зрения для детекции и сегментации товаров на сортировочной ленте. В ней рассматриваются методы повышения точности детектирования объектов, разрабатывается модуль компьютерного зрения для решения задач складской логистики, включая детекцию и сегментацию, а также создается синтетическая тестовая среда для проверки разработанных методов.

## **1 Обзор текущих производственных решений и способы их оптимизации.**

### **Анализ альтернативных способов определения объемно весовых характеристик**

В этой главе рассматривается обзор текущих производственных решений в сфере логистики, а именно автоматизация измерений и учета на складе (ОВХ — объемно весовые характеристики). Обзор популярных технологий. Определение слабых и сильных сторон текущего производственного решения. Обзор альтернативных методов автоматизации измерений объемно весовых характеристик. Выявления сильных и слабых сторон каждой рассматриваемой технологии. Составление сравнительной характеристики.

В настоящее время множество процессов подвержены автоматизации. Особенно успешно автоматизация рутинных и технически сложных процессов внедряется в производство. С каждым годом все больше предприятий внедряет и разрабатывает новые способы автоматизации производственных процессов. Автоматизация производственных процессов повышает эффективность всего предприятия в различных направлениях. Позволяет повысить качество конкретного процесса, уменьшить количество ошибок, вызванных человеческим фактором, уменьшить количество брака. Автоматизация позволяет уменьшить количество различных ресурсов (технических, человеческих, экономических) требуемых для конкретного процесса, следовательно освободившиеся ресурсы можно применить для масштабирования производства и т. п.

Наиболее актуальна автоматизация складской логистики и многие крупные компании давно этим занимаются. В рамках склада существует множество процессов, которые можно оптимизировать и автоматизировать при помощи технических средств. На складе каждой компании можно встретить различные технические решения автоматизации.

В этой главе обзореваются возможные способы автоматизации складской логистики, а именно автоматизация определения объемно весовых характеристик товара на складе.

## 1.1 Анализ предметной области

### 1.1.1 Устройство современных складов

Устройство склада имеет множество автоматизированных и ручных процессов. Современный склад представляет из себя набор технических средств и технологий, нацеленных на максимальную производительность обработки товара и посылок. Современные складские помещения — это полноценная киберфизическая система с возможностью вертикального и горизонтального масштабирования. Как правило склады занимают огромные площади и обеспечить нужный уровень коммуникации на большом количестве устройств — это тоже сложная задача и нужно тоже это учитывать при создании концепций.

Склад представляет собой совокупность систем, нацеленных на обработку товара с момента получения на склад, до момента отправки со склада.

На современных складах используется большое разнообразие транспортных и сортировочных лент, сканеров и интерфейсов их взаимодействия друг с другом.

На рисунке 1 представлена пример транспортных лент в действующем складском помещении.



Рисунок 1 – Складское помещение

### 1.1.2 Движение посылки внутри склада

Движение посылок внутри склада осуществляется посредством перемещения на технике, конвейерах и подвижных лентах. Существует множество вариаций статусной модели, рассмотрим наиболее общую модель и посмотрим на путь товара внутри склада с момента поступления до момента выхода со склада. Данная описанная статусная модель не имеет большой декомпозиции по причине того, что с момента поступления товара до момента выезда происходит множество длительных и коротких процессов и описывать все действия не имеет смысла. Также в различных предприятиях, процессы могут иметь отличную друг от друга очередность и вариативность.

На рисунке 2 изображена общая схема движений товара в складской логистике. В процесс «Обработка товара на складе» поступает товар из внешнего источника, под внешним источником подразумевается любой способ трансфера товара до склада и так же из него уходит товар во внешний источник, который тоже является вариантом трансфера до следующего склада, пункта выдачи или непосредственно до заказчика.

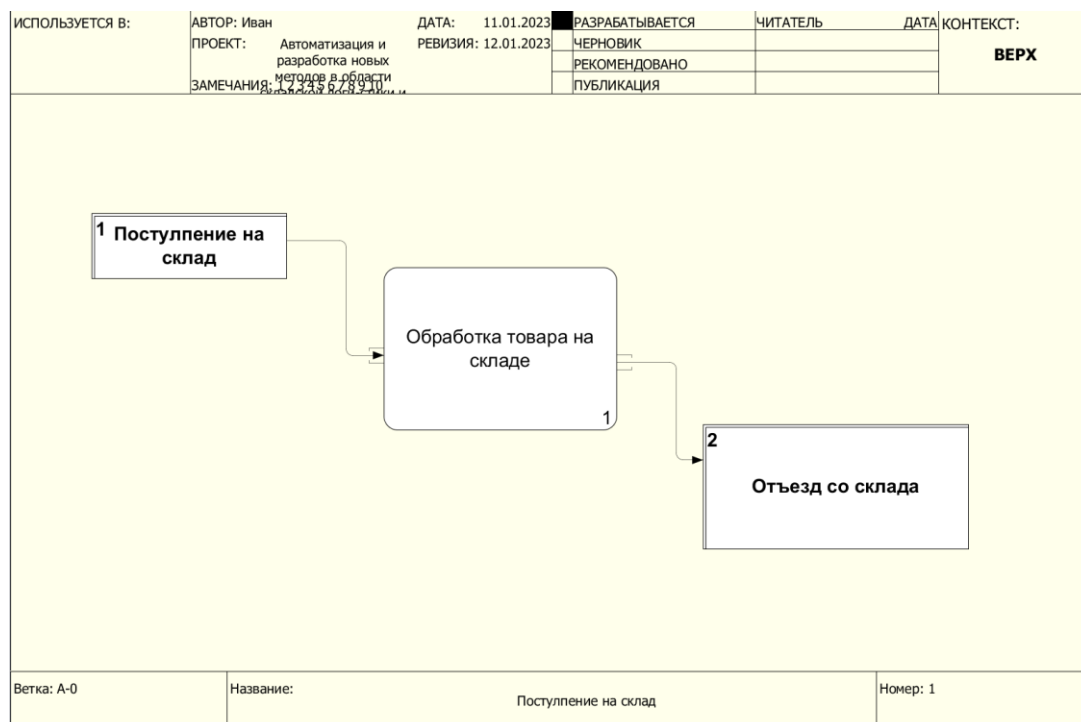


Рисунок 2– Схема движения груза

На рисунке 3 изображена декомпозированная схема «Обработка товара на складе».

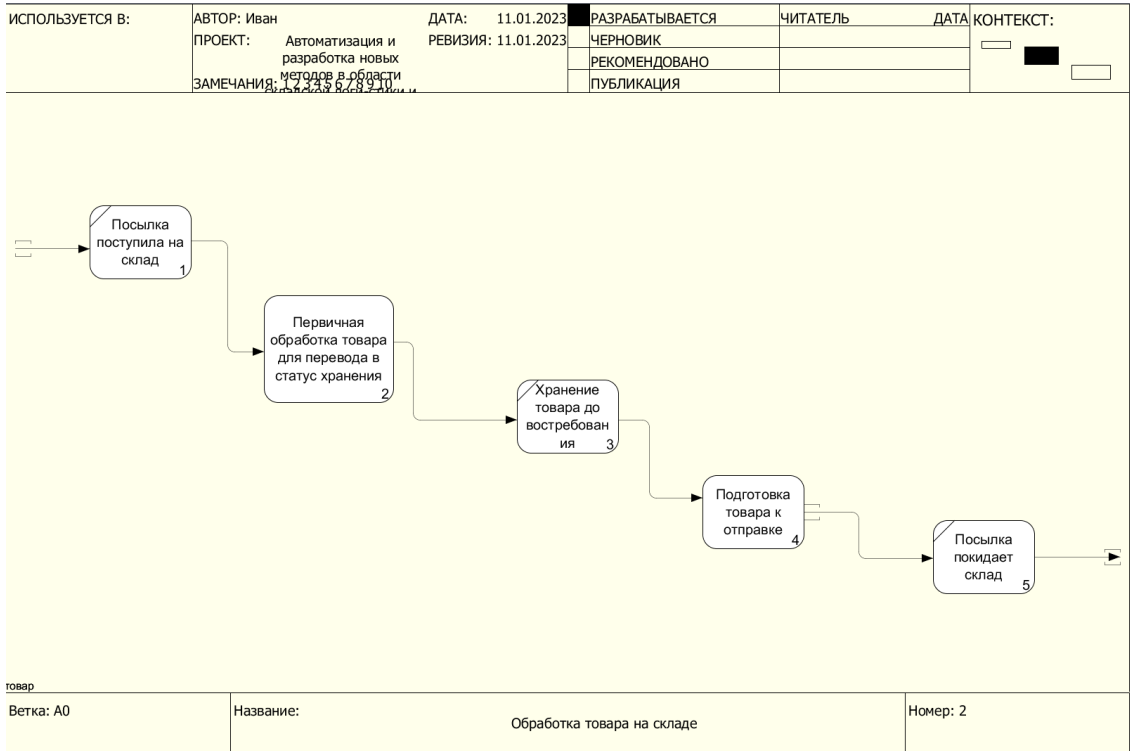


Рисунок 3 – Декомпозиция процесса «Обработка товара на складе»

На рисунке 4 изображена декомпозированная схема «Первичная обработка товара».

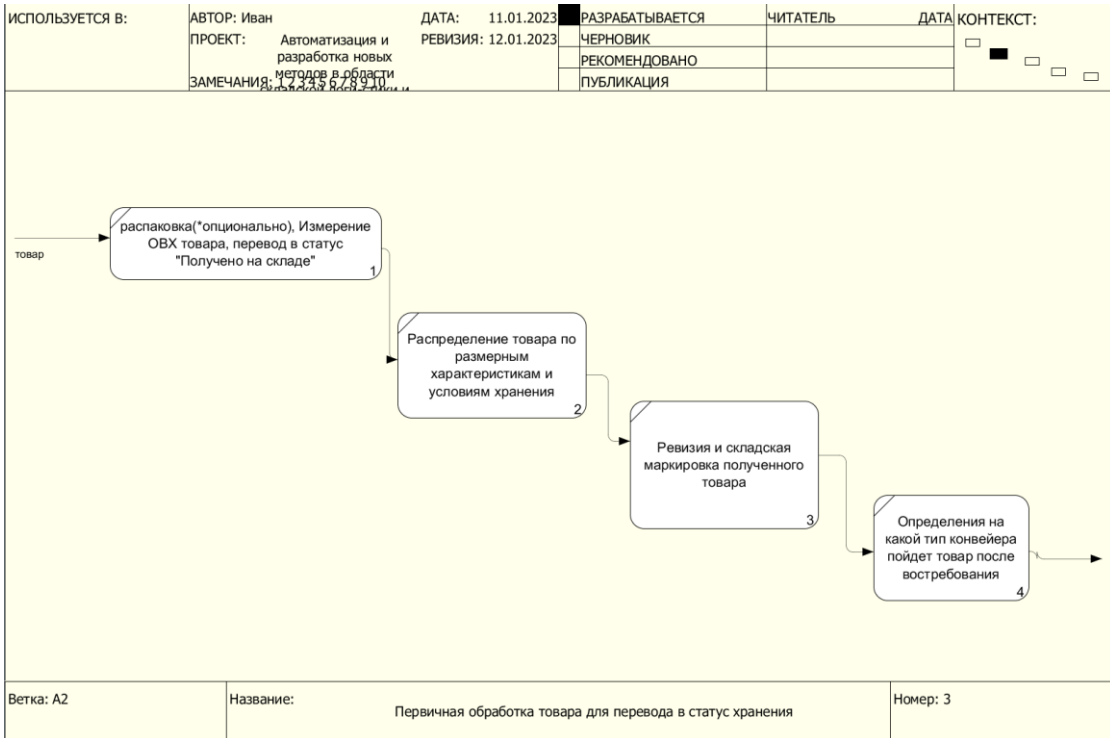


Рисунок 4 – Декомпозиция процесса «Первичная обработка товара»

На рисунке 5 изображена декомпозированная схема «Подготовка товара к отправлению».

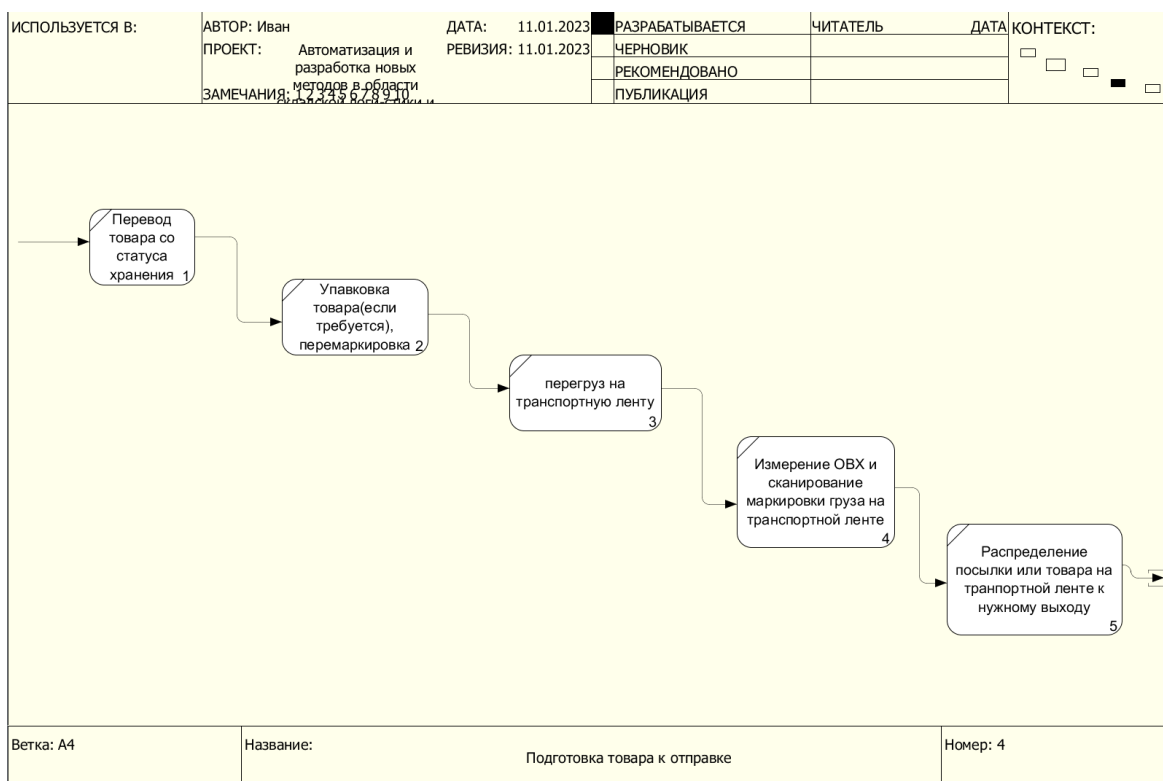


Рисунок 5 – Декомпозиция процесса «Подготовка товара к отправлению»

### 1.1.3 Текущие технологические решения складской логистики

Для оптимизации складской логистики используются различные технические средства позволяющие вести учет отправок и сортировку товара. Учет товара происходит посредством считывания штрихкодов и меток на товаре во время транспортировки по ленте, так же существует процесс определения габаритных и весовых характеристик упакованного или товара для определения дальнейшей судьбы товара и направления его на нужный выход. На рисунке 6 представлено действующее решение сканера на транспортировочной ленте.

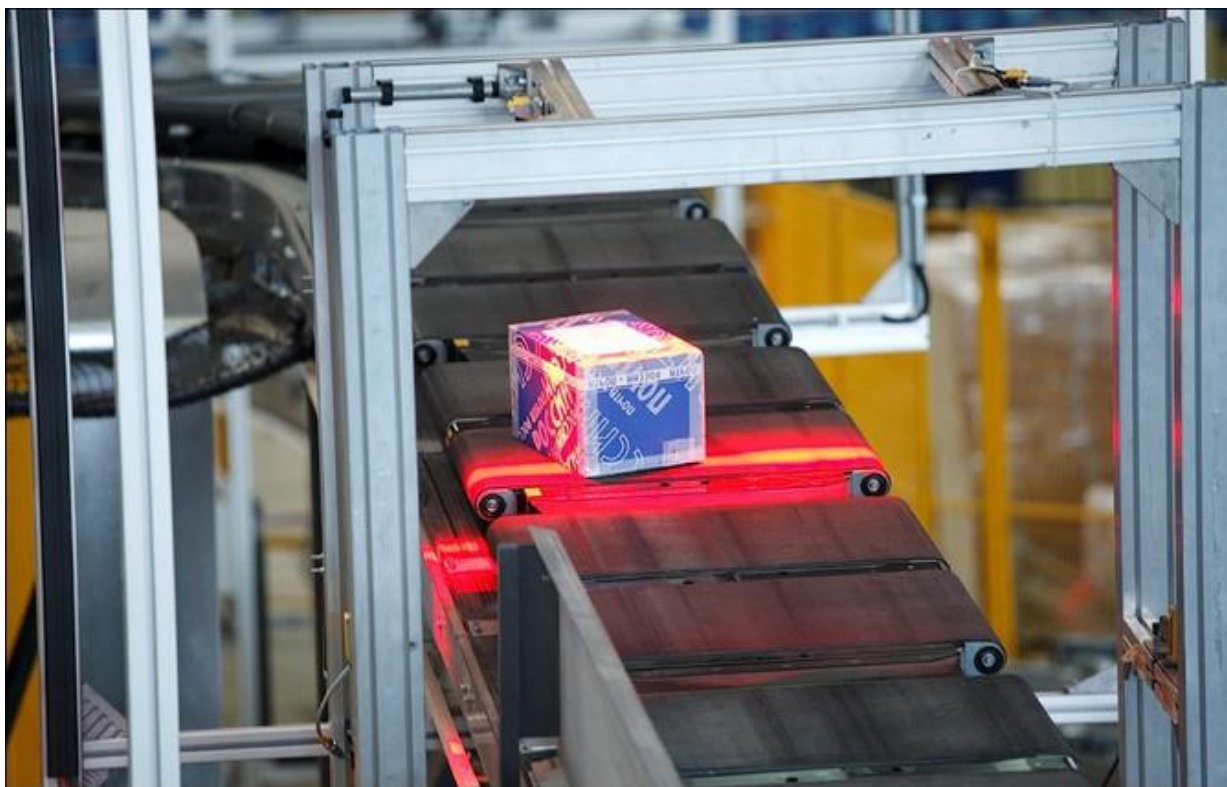


Рисунок 6 – Сканер штрихкодов на подвижной ленте

Сортировка и транспортировка до нужного выхода может направляться посредством башмачной ленты.

Определение весогабаритных характеристик и идентификации маркировки упакованного товара занимаются сортеры, различного рода сканера, динамические весы. На рисунке 7 представлено сортировочная лента, оснащенная «башмаками» для перевода груза в нужное ответвление транспортной сети.





Рисунок 7 – Сортировочная лента, оснащенная «башмаками»

## **1.2 Формулирование проблем связанных с определением весогабаритных характеристик**

Наиболее актуальной проблемой процесса определения весогабаритных характеристик является точность измерений и производительность.

Точность измерений влияет на заявленный объем готового к отправке товара и в следствии могут возникнуть сложности во время перевода в транспортировку, товар попросту может не поместиться в предполагаемую машину и превысить допустимые нормы веса.

Транспорт, уходящий со склада, должен быть максимально оптимально заполнен, но при этом не превышать стандарты безопасности. Для этого предварительно перед отправкой происходит учет загруженного товара и предварительное моделирования какую упаковку и товар допустимо отправить на конкретном виде транспорта. И весь этот процесс происходит в многопоточном и параллельном формате, поэтому требуется максимально быстро и точно произвести измерения и как можно скорее внести информацию в базу.

В различных компаниях измерения могут проходить на разных этапах обработки груза и различными способами в том числе ручным. Ручной способ измерения весогабаритных характеристик имеет самую низкую точность и производительность и может часто быть источником проблем на последующих этапах.

Существуют производственные решения, которые позволяют измерять грузы в определённом весоразмерном диапазоне в полуавтоматическом режиме. Такой способ имеет большую точность и время обработки одного груза сводится к сумме времени на считывания штрихкода и постановки груза на стенд.

На рисунках 8 и 9 изображены производственные решения для определения габаритов груза и веса различных формфакторов.



Рисунок 8 – Устройство для определения габаритов и веса



Рисунок 9 – Устройство для определения габаритов и веса грузов произвольной формы

Также в крупных компаниях используются способы динамического измерения габаритов груза непосредственно на транспортировочной ленте. В этом случае, когда посылка входит в зону измерений, считывается ее штрихкод специальным сканером который способен произвести считывание с любой грани груза и далее производятся измерения. Следующим этапом данные отправляются в базу, где происходят дальнейшие операции по отправке груза.

Динамическое измерение габаритов является самым эффективным решением при должном уровне точности.

Таким образом подводя промежуточный итог, зафиксируем что главные требования к способу определения весогабаритных характеристик это:

- точность измерений;
- скорость измерений;
- минимизация нештатных поведений;
- бесперебойность;
- доступность к внедрению.

### **1.3 Технические способы определения весогабаритных характеристик**

Для начала анализа способов определения габаритов груза, требуется определиться с его максимальными и минимальными размерами, а также с его геометрической формой.

Посылки бывают разные по размеру и форме. И при каких-то размерах произвести измерения намного эффективнее вручную, например на грузе, который не помещается на транспортную ленту. Так же можно пренебречь сверхмалыми посылками, которые практически ничего не весят и почти не занимают места (конверты, мелкие пакеты), погрузка таких посылок для большого количество подобных посылок предположительно должна быть в отдельных агрегируемых коробках.

Упакованные товары чаще всего геометрически представляют из себя скруглённый, криволинейный параллелепипед. Для удобства учета и подсчета примерного объема груза предложим трехмерную систему измерения, длину, ширину, высоту и все криволинейные фигуры груза будем сводить к идеальному параллелепипеду.

Существует множество датчиков и алгоритмов, которые позволяют определить габариты движущегося с постоянной скоростью объекта. Рассмотрим наиболее полярные и очевидные.

#### **1.3.1 Датчики расстояния**

Наиболее простые для использования и точности является использование датчиков расстояния. Датчики расстояния позволяют определять расстояния и

на основе данных путем нехитрых манипуляций можно узнать размер каждой проекции посылки, но это при условии, что груз представлен в виде короба.

Существует множество вариаций использования данных датчиков, оптимально использование их в двухсторонней рамке. В каждый момент времени движения посылки будет составлять трехмерная карта проходящего через него объекта и на основе полученных данных будет строиться модель посылки и рассчитываться показатели длины, ширины, высоты, это при условии, что мы знаем скорость движения сортировочной ленты.

На рисунке 10 представлено устройство позволяющие считывать штрих-коды с любой грани груза и измерять его весогабаритные характеристики.

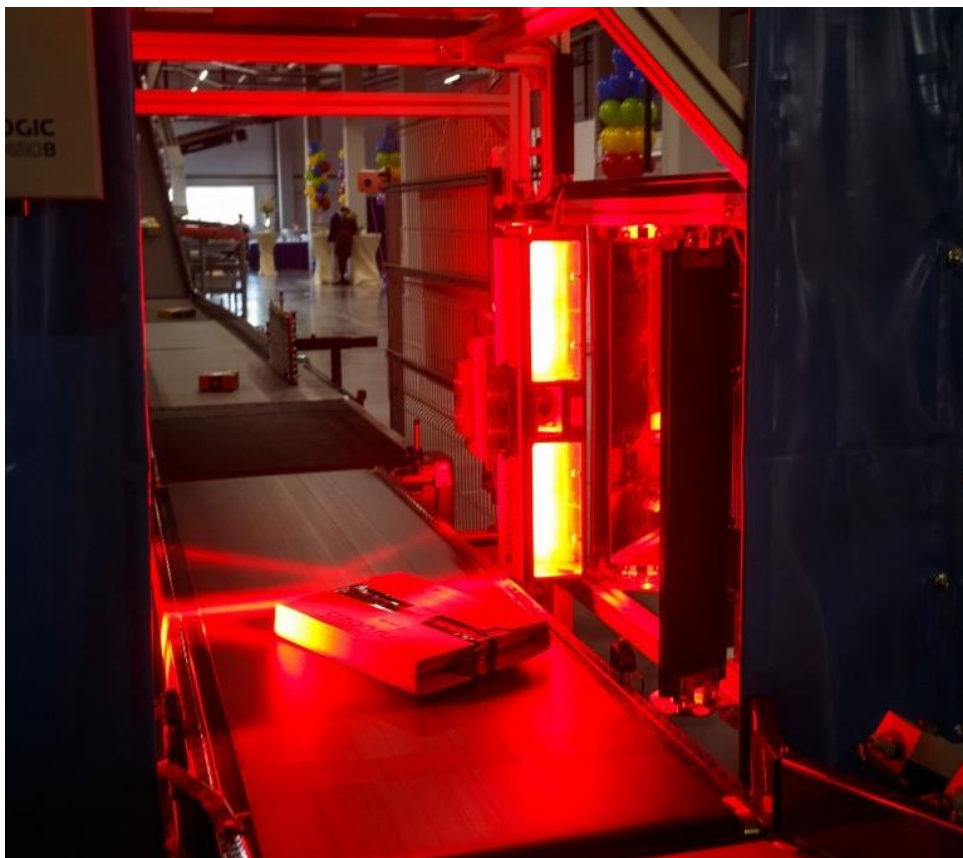


Рисунок 10 – Сканер с возможностью определения габаритов и веса

Существует несколько типов датчиков расстояния, самые популярные это ультразвуковые и лазерные.

#### 1.3.1.1 Лазерный датчик расстояния

Лазерный датчик расстояния — это электронное устройство, использующие свойства лазерного излучения для измерения расстояний до поверхностей.

Лазерный датчик направляет прямой луч и замеряет расстояние до конкретной точки. Его достоинства:

- высокая точность измерений  $\pm 1-5$  мм на всем диапазоне;
- измерение расстояний даже до очень маленьких объектов.

Недостатки:

- некорректная работа при ярком освещении и в солнечную погоду. (проблема решается использованием специальных отражателей);
- невозможность измерить расстояние отражающих поверхностей.

На рисунке 11 изображена схема работы лазерного датчика расстояния.

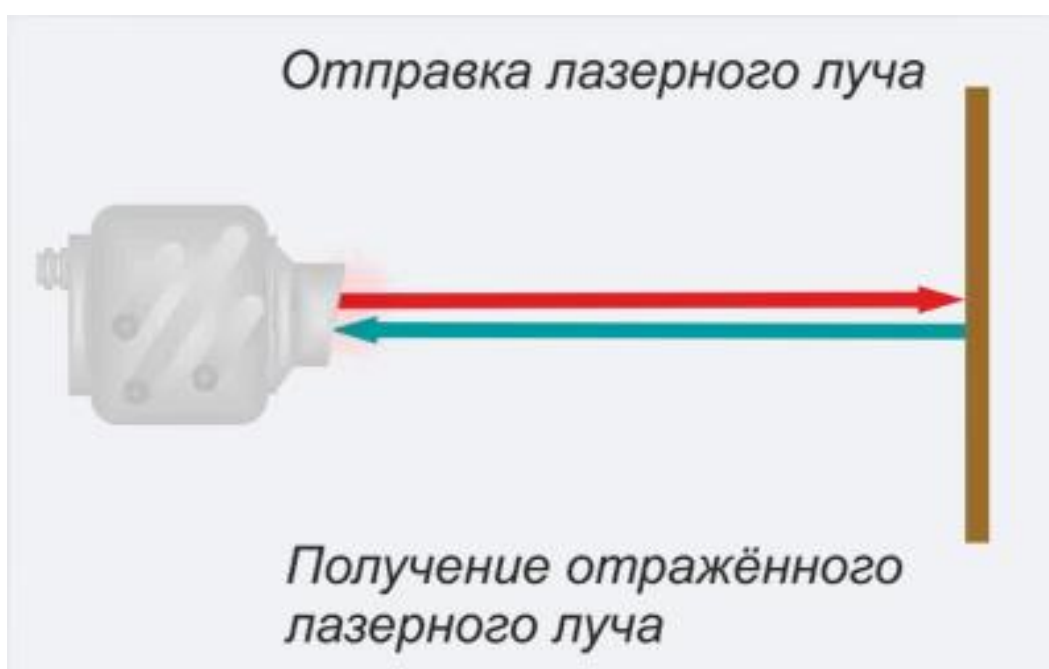


Рисунок 11– Схема работы лазерного датчика

#### 1.3.1.2 Ультразвуковой датчик расстояния

Ультразвуковой датчик применяется для обнаружения и определения расстояния до объекта, а также для контроля их движения. Передатчик излучает звуковые колебания, частота которых превышает 20 кГц.

Ультразвуковой датчик направляет на объект ультразвуковую волну, которая сама по себе делает более широкий захват предмета.

Принцип работы ультразвукового дальномера схож с эхолотом и основан на отражении звука от измеряемого объекта. Основным преимуществом таких устройств являются:

- измерение расстояний до объектов, пропускающих свет;
- невысокая стоимость.

Недостатки:

- диапазон работы прибора ограничен из-за рассеивания звуковых волн;
- сравнительно невысокая точность измерений;
- объект измерений должен быть достаточно большим, а его поверхность не должна поглощать звук.

На рисунке 12 изображена схема работы лазерного датчика расстояния.

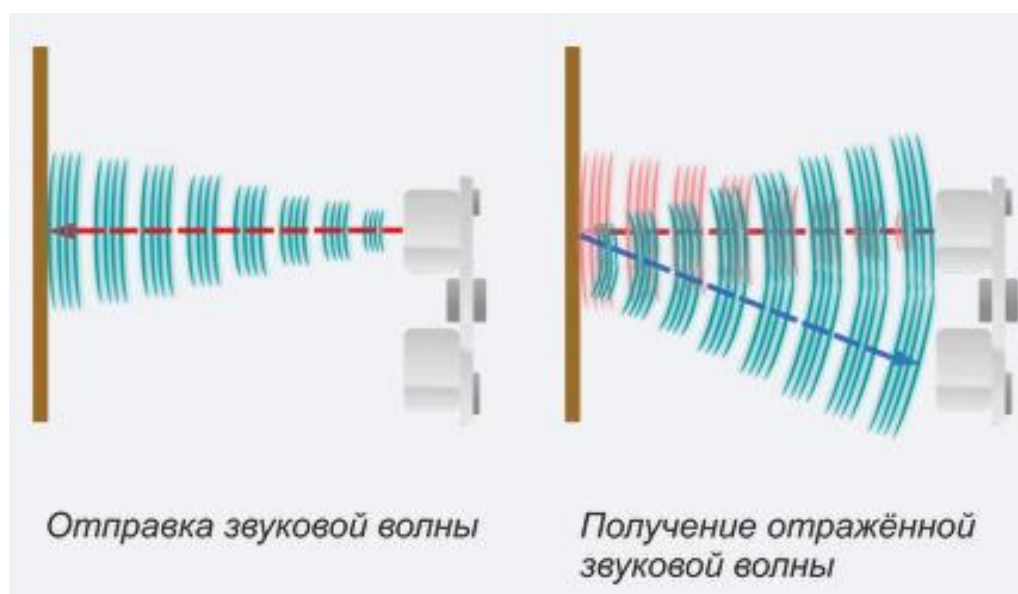


Рисунок 12 – Схема работы ультразвукового датчика

### 1.3.2 Техническое зрение

Применение технологий компьютерного зрения являются наиболее технологичными и сложными с точки зрения разработки, но при должном обучении будут показывать точные результаты. Существует несколько подходов использования камеры для определения габаритных характеристик посылок.

#### 1.3.2.1 Компьютерное зрение

В данном подходе предполагается использование одной камеры, расположенной так что при определенном ракурсе просматривалось сразу три грани посылки или предположительные грани, очень важен цвет самой посылки так как



он должен контрастировать на фоне транспортной ленты и окружающего пространства. Это очень важный факт для корректного определения контуров посылки и дальнейшего его детектирования.

На рисунке 13 слева представлено изображение, подготовленное для определения контуров, справа, исходное изображение.

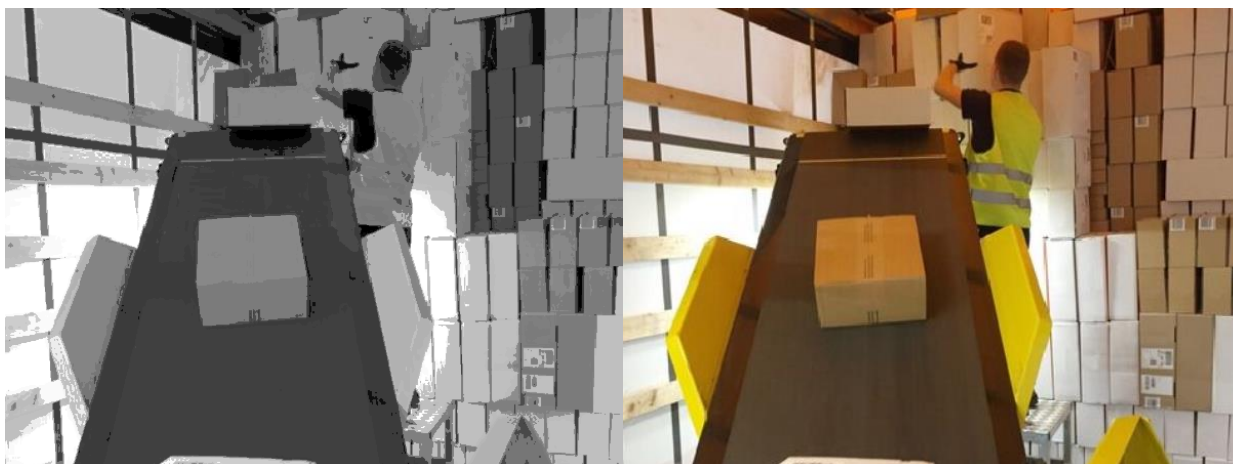


Рисунок 13 – Считывание контуров с фотографии

Габариты могут определяться покадрово или в режиме детектирования объектов в потоковом видео. Камера обязательно должна быть откалибрована и на краях ленты в зоне детектирования должны быть расположены маркеры, которые позволят определить расстояния до объекта и размер проекций.

На рисунке 14 представлен пример детекции граней груза.



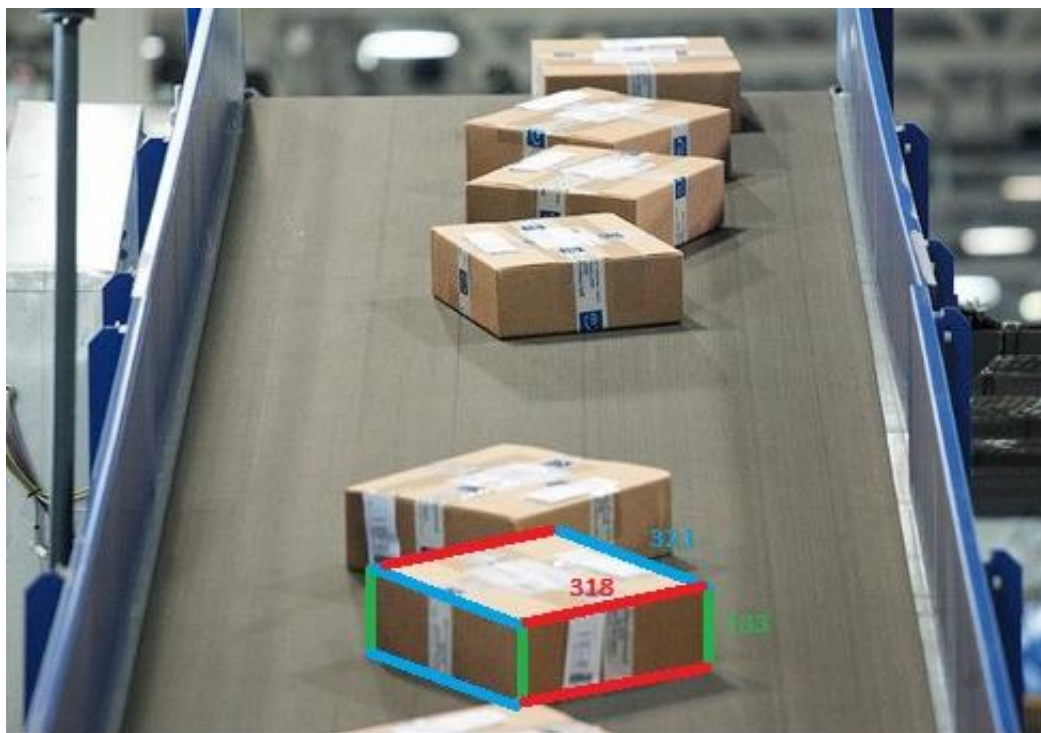


Рисунок 14 – Схематическое представление работы технического зрения

#### 1.3.2.2 Стереозрение с проекцией калибровочного поля

Существует более усовершенствованный способ определения габаритных характеристик.

В основе этого метода является стереозрение. Подобная технология используется при 3D-сканировании в аддитивных технологиях. На одной планке на определённом расстоянии друг от друга расположены две камеры. Вся установка откалибрована на определённый участок подвижной ленты. Проектор подсвечивает определённым контрастным узором посылку и транспортную ленту. Камеры делают снимки, далее производится сопоставление изображений с разных камер и на основе данных преломления узора, строится частичная модель посылки. Далее в постобработке она автоматически достраивается и вычисляются ее габаритные характеристики.

На рисунке 15 представлен 3d-сканер для создания 3d-моделей физических объектов.



Рисунок 15 – 3d-сканер

На рисунке 16 изображен вариант подсветки для считывания искажений.

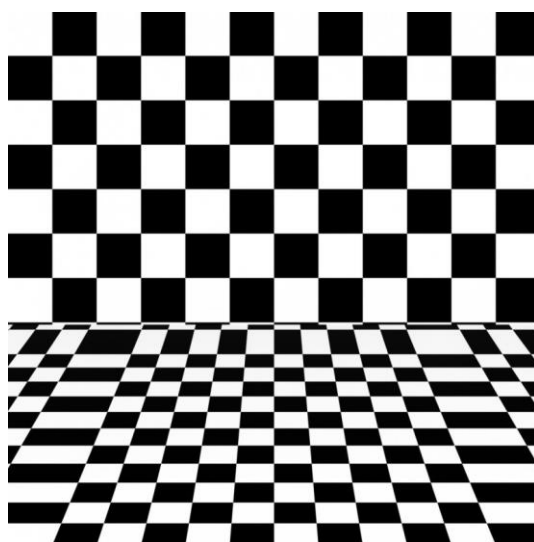


Рисунок 16 – Вариант подсветки для считывания искажений

### 1.3.3 Лидар

Лидар (LiDAR) — это технология дистанционного зондирования, которая использует лазерный импульс для сбора измерений, которые затем можно использовать для создания 3D-моделей, карт объектов и окружающей среды.

LiDAR является аббревиатурой от Light Detection and Ranging - обнаружение света и дальности (расстояния).

Он работает аналогично радару и сонару, но использует световые волны от лазера вместо радио или звуковых волн. Система лидар рассчитывает, сколько времени требуется свету, для попадания в объект и отражения обратно в сканер. Расстояние рассчитывается с использованием скорости света.

На рисунке 17 изображена глубинное изображение, полученное технологией LiDAR.

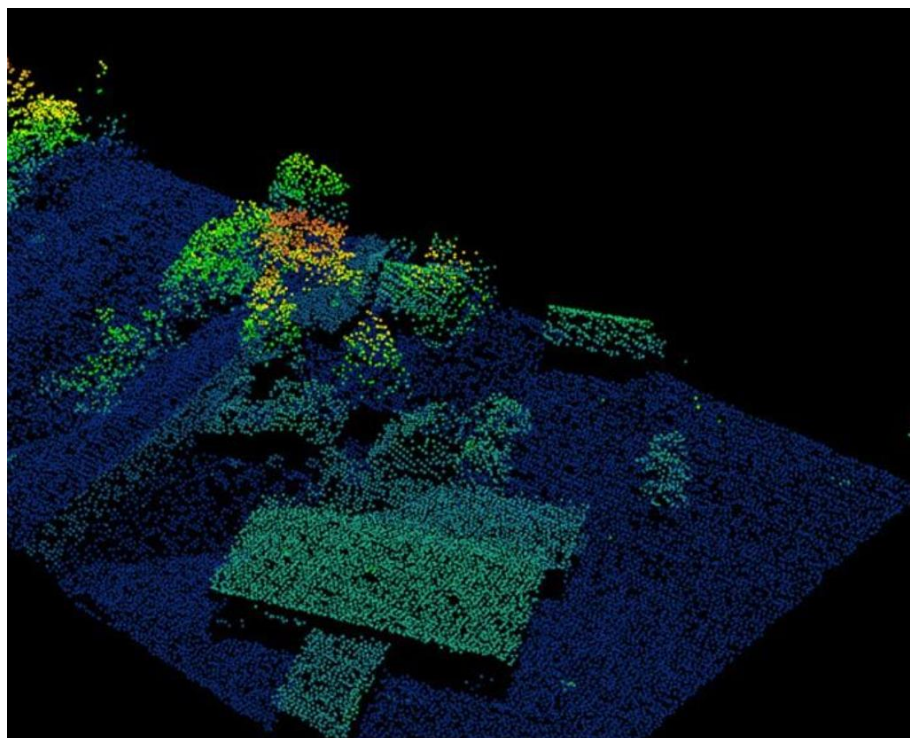


Рисунок 17 – Карта глубины

Системы могут генерировать около 1 000 000 импульсов в секунду. Каждое из этих измерений или результатов может затем быть преобразовано в трехмерную визуализацию, которая представляет собой облако точек.

Технология Lidar позволяет собирать трехмерные измерения. Использование данной технологии также наблюдается повышенный уровень внедрения приложений автоматизации.

Достоинства технологии Lidar:

- высокая скорость и точность сбора;
- высокое проникновение;
- не зависит от интенсивности света в окружающей среде и может использоваться ночью или на солнце;

- высокое разрешение изображения по сравнению с другими методами;
- отсутствие геометрических искажений;
- легко интегрируется с другими методами сбора;
- ЛИДАР имеет минимальную зависимость от человека, что хорошо в определенных сферах, где человеческая ошибка стоит дорого.

Недостатки технологии Lidar:

- стоимость лидара достаточно высока;
- системы LIDAR плохо работают в условиях сильного дождя, тумана или снега;
- системы LIDAR генерируют большие наборы материалов, которые требуют больших вычислительных ресурсов для обработки;
- ненадежные результаты в турбулентных условиях;
- в зависимости от принятой длины волны рабочие характеристики систем ограничены по высоте, поскольку импульсы, генерируемые в определенных типах, становятся неэффективными на определенных высотах.

Технология Lidar в рамках задачи определения габаритов может с большой точностью детектировать грузы во движении по ленте и при помощи глубинного зрения формировать трехмерную модель объекта, определяя его длину, ширину и высоту и мгновенно передает показатели в систему управления складом.

#### **1.4 Выбор технологии для дальнейшего исследования**

У каждой технологии есть свои сильные и слабые стороны. Каждую из технологий можно использовать в задаче определения весогабаритных характеристик груза, но требуется понимать в каком контексте их использовать.

В случае если требуется определять весогабаритные характеристики груза во время движения по ленте, то лучшими вариантами будут технологии Lidar, стереозрения, характеристика представлена в Таблице 1.

Если требуется статичное определение габаритов груза в коробках или приближенных к форме параллелепипеда, то наиболее простым, надежным и дешевым решением будут являться лазерные и ультразвуковые датчики.

Для статичного определения габаритов груза со сложной геометрией будут являться технологии стереозрения.

Наиболее эффективно будет использовать сразу несколько технологий для уменьшения общей погрешности погрешности и повышению адаптивности.

Также стоит учитывать свойства материала упаковки для каждого из способов.

Таблица 1 – Характеристика рассматриваемых технологий

	Измерение в динамике	Измерений в статике	Измерение простых форм (короб, близкие к параллелепипеду)	Измерение сложных форм	Ограничения по материалу упаковки	Стоимость	Сложность разработки
Ультразвуковой датчик	Эффективность измерения в динамике стремится к нулю	Идеально подходит для измерений в статике грузов определенной формы	Поддерживает	Не поддерживает	Материал упаковки и изделия не должен поглощать шум	Самое дешевое решение, от 1\$, но для нормальных показаний требуется более дорогая модель	Простой
Лазерный датчик расстояния	Использование для измерений в динамике допускается только в совокупности с другой технологией	Идеально подходит для измерений в статике грузов определенной формы	Поддерживает	Не поддерживает	Материал не должен быть зеркальным	от 2\$, но для нормальных показаний требуется более дорогая модель	Простой
Компьютерное зрение (одна камера)	Допускается измерение в динамике при соблюдении условий к форме и материалу груза. Оценочная точность приемлемая	Допускается измерение при соблюдении условий	Поддерживает	Поддерживает	Цвета должна контрастировать на фоне	Приемлемые для работы от 20\$	Сложный
Стереозрение	Точные динамические измерения	Идеально подходит для измерений в статике грузов всех форм	Поддерживает	Поддерживает	Цвета должна контрастировать на фоне	Требуется минимум две камеры о 20\$ и желательно проектор от 30\$	Средний
Лидар	Наиболее точные измерения	Подходит для измерений в статике грузов всех форм, но не является оптимальным решением из-за стоимости	Поддерживает	Поддерживает	Материал не должен быть зеркальным	75\$ за самый дешевый вариант с большим количеством ограничений	Средний

## **1.5 Вывод к главе**

В рамках данной главы произведен обзор текущих производственных решений в сфере логистики, а именно автоматизация измерений и учета товаров при получении на склад (ОВХ — объемно весовые характеристики).

Произведён обзор технологий позволяющих автоматически определять весогабаритных характеристики и выявление их слабых и сильных сторон их достоинств и недостатков.

На основе изученного материала были получены следующие результаты:

- Изучены технологии позволяющие определять весогабаритных характеристики грузов на складе.
- Произведен обзор текущих производственных решений определения весогабаритных характеристик грузов.
- Произведена сравнительная характеристика каждой из технологий.
- Выявлены требования к будущему технологическому решению.
- Выбран набор самых оптимальных технологий для решения прикладной задачи.

## **2 Использование глубокого обучения и нейронных сетей для автоматического распознавания и детекции товаров на сортировочной ленте.**

Целью данной главы является исследование методов и инструментов для автоматизации учета товаров на складе, а также оценка эффективности их применения. Основное внимание будет уделено применению глубокого обучения и нейронных сетей для автоматического распознавания товаров на сортировочной ленте. Путем применения современных методов машинного обучения и анализа данных можно достичь точности и скорости учета товаров, которые трудно достичь с использованием классических ручных методов.

Дальнейшее внедрение современных технологий и инноваций может значительно снизить издержки и увеличить эффективность складской логистики и доставки. Это также даст предприятиям возможность оперативно реагировать на изменения рыночных условий и обеспечивать высокое качество обслуживания клиентов. Складская логистика и доставка является важным аспектом в современных бизнес-процессах, особенно в контексте электронной коммерции и растущего объема перевозок. Одним из ключевых аспектов эффективной работы склада является точный и оперативный учет поступающих товаров. Однако ручные методы учета и измерения товаров на складе могут быть трудоемкими и подвержены ошибкам, особенно при больших объемах поступления товаров.

С появлением современных технологий, таких как глубокое обучение и нейронные сети, открываются новые возможности для улучшения процессов складской логистики и доставки. Автоматическое распознавание и измерение товаров на сортировочной ленте с использованием этих методов позволяет сократить время и повысить точность учета, что в свою очередь способствует повышению операционной эффективности и улучшению общей качества обслуживания.



## **2.1 Анализ применения глубокого обучения и нейронных сетей в области складской логистики**

Анализ применения глубокого обучения и нейронных сетей в области складской логистики является важной частью исследования, поскольку это технологии, которые могут значительно улучшить процессы складской логистики и доставки. Рассмотрим некоторые из аспектов применения глубокого обучения и нейронных сетей в этой области.

### **2.1.1 Формулирование сценариев применения глубокого обучения и нейронных сетей в области складской логистики**

Наиболее актуальными сценариями использования моделей нейронных сетей в контексте складской логистики являются:

- 1) Распознавание товаров. Глубокое обучение и нейронные сети показали высокую эффективность в задачах классификации и распознавания объектов на изображениях. В контексте складской логистики они могут быть использованы для автоматического распознавания и классификации товаров на сортировочной ленте. Модели глубокого обучения, такие как сверточные нейронные сети (CNN), могут обучаться на больших наборах данных и способны точно идентифицировать различные типы товаров на основе их визуальных характеристик.
- 2) Измерение товаров. С помощью методов машинного обучения можно разработать системы, которые могут оценивать размеры и габариты товаров на основе данных полученных в результате сегментации изображений. Это позволяет автоматизировать процесс измерения товаров, уменьшить вероятность ошибок и повысить точность учета.
- 3) Обнаружение дефектов. Путем обучения моделей на размеченных данных можно создать систему, способную автоматически выявлять дефекты, такие как трещины, сколы, царапины и прочие повреждения, на поверхности товаров. Это позволяет оперативно выявлять и изолировать поврежденные товары, улучшая качество контроля качества на складе.

### 2.1.2 Сведение сценариев применения глубокого обучения и нейронных сетей к классическим задачам компьютерного зрения

Сформулированные сценарии и варианты использования можно свести к конкретным классическим задачам компьютерного зрения таким как:

- распознавание объектов – задача состоит в определении и классификации объектов на изображении или видео. Это может быть распознавание товаров, людей, транспортных средств, животных, предметов и т. д.;
- детектирование объектов – задача заключается в определении и локализации объектов на изображении или видео. Это может включать поиск лиц, автомобилей, пешеходов и других объектов;
- сегментация изображений – задача состоит в разделении изображения на семантические или пиксельные области. Это позволяет выделить объекты и их контуры на изображении;
- оптическое распознавание символов – задача заключается в распознавании текста на изображении или видео. Это может использоваться для автоматического чтения текста с документов, этикеток, номеров автомобильных номеров, табличек и т. д.;
- отслеживание объектов – задача заключается в отслеживании движущихся объектов на видео и определении их траектории. Это может использоваться, например, в системах видеонаблюдения или автоматическом вождении;
- реконструкция трехмерных моделей – задача состоит в восстановлении трехмерной модели объекта или сцены на основе двумерных изображений или видео. Это может использоваться в виртуальной реальности, архитектурном проектировании, медицинской диагностике и других областях.

## 2.2 Применение глубокого обучения для решения задач складской логистики

### 2.2.1 Модель компьютерного зрения Yolo

Yolo – это одна из наиболее известных моделей для обнаружения объектов на изображениях в реальном времени. Она основана на сверточных нейронных сетях и позволяет достичь высокой скорости обработки без ущерба точности. YOLO разделяет изображение на сетку ячеек, и каждая ячейка предсказывает границы и классы объектов, содержащихся внутри нее.

В рамках алгоритма YOLO исходное изображение сначала разбивается на сетку из  $N \times N$  ячеек. Если центр объекта попадает внутрь координат ячейки, то эта ячейка считается ответственной за определение параметров местонахождения объекта. Каждая ячейка описывает несколько вариантов местоположения ограничивающих рамок для одного и того же объекта. Каждый из этих вариантов характеризуется пятью значениями — координатами центра ограничивающей рамки, его шириной и высотой, а также степени уверенности в том, что ограничивающая рамка содержит в себе объект. Также необходимо для каждой пары класса объектов и ячейки определить вероятность того, что ячейка содержит в себе объект этого класса. Таким образом, последний слой сети, принимающий конечное решение об ограничивающих рамках и классификации объектов работает с тензором размерности  $N \times N \times (5B + C)$ , где  $B$  — количество предсказываемых ограничивающих рамок для ячейки,  $C$  — количество классов объектов, определённых изначально.

Архитектура YOLOv7 включает в себя несколько улучшений по сравнению с предыдущими версиями YOLO, такими как использование более эффективной магистральной сети, использование блоков для повышения точности и использование многомасштабного прогнозирования для лучшей обработки объектов разных размеров. Алгоритм также позволяет выбирать из нескольких базовых весов для улучшения компромисса между скоростью и точностью.

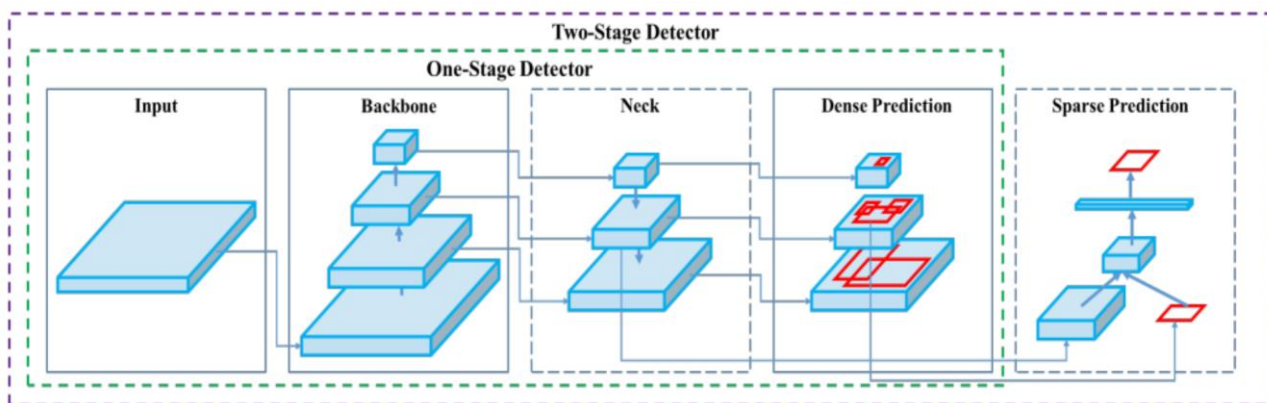


Рисунок 18 – Архитектура YOLO

Алгоритм YOLO работает быстрее алгоритмов семейства R-CNN за счёт того, что поддерживает дробление на константное количество ячеек вместо того, чтобы предлагать регионы и рассчитывать решение для каждого региона отдельно, однако, в качестве проблем YOLO указывается плохое качество распознавания объектов сложной формы или группы небольших объектов из-за ограниченного числа кандидатов для ограничивающих рамок.

В целом, YOLO— это мощный и быстрый алгоритм обнаружения объектов, который может быть использован в широком спектре приложений, включая самоуправляемые автомобили, системы наблюдения и робототехнику.

### 2.2.2 Обучение модели Yolo на пользовательском наборе данных OpenImages

Для дообучения модели Yolo в контексте складской логистики, используем пользовательский набор данных OpenImages.

#### 2.2.2.1 Набор данных OpenImages

OpenImages — это открытый набор данных, содержащий миллионы помеченных изображений, собранных через платформу Google Images. Он предназначен для обучения алгоритмов компьютерного зрения и решения различных задач обработки изображений.

Набор данных включает разнообразные категории объектов, каждый из которых классифицируется и помечается на изображениях. Каждое изображение снабжено метаданными, такими как метки классов, границы объектов и другая сопутствующая информация.

OpenImages является одним из крупнейших и наиболее разнообразных наборов данных с разметкой объектов, доступных для общего использования. Он предоставляет реалистичные изображения, собранные из поисковой системы Google Images, что обеспечивает широкое покрытие объектов различных классов и сцен.

Набор данных OpenImages имеет несколько версий с различными подмножествами и вариантами разметки. Например, последняя версия V7 содержит более 9 миллионов изображений, помеченных более чем 6000 классами объектов. Этот набор данных постоянно обновляется и поддерживается сообществом.

Набор данных OpenImages подходит для различных задач, таких как классификация, детектирование объектов и сегментация изображений, и является ценным ресурсом для исследователей и разработчиков, интересующихся обработкой изображений и компьютерным зрением.



Рисунок 19 – Дендограмма объектов датасета OpenImages

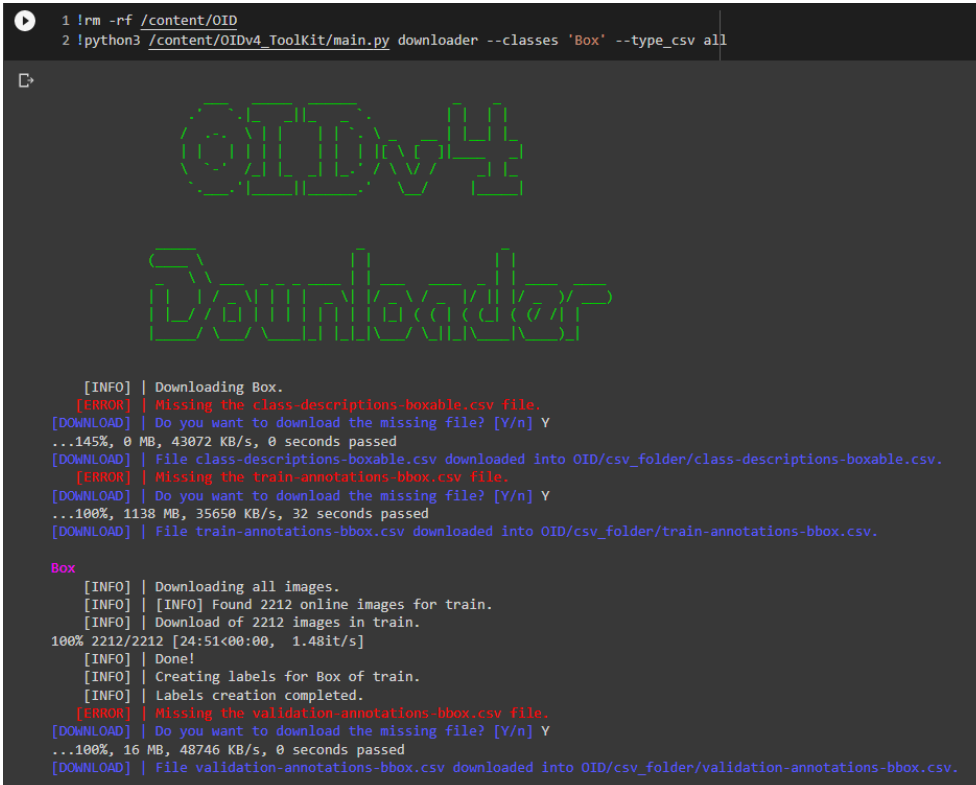
### 2.2.2.2 Подготовка входных данных для модели Yolo

Весь набор данных OpenImages состоит из миллионов изображений. Общий объем данных OpenImages может занимать несколько сотен терабайт. Однако, точный размер всего набора данных может варьироваться в зависимости от версии и состава данных, а также от способа хранения и сжатия изображений.

Учитывая большой объем данных OpenImages, в контексте нашей задачи требуется использовать только подмножество данных, содержащее информацию о нескольких объектах. Это позволит уменьшить объем данных, которые нужно обрабатывать и хранить, и сосредоточиться только на интересующих классах объектов.

Для загрузки нужного подмножества используем утилиту OIDv4 ToolKit.

Open Images Dataset v4 ToolKit – это инструмент для работы с набором данных OpenImages.



```
1 !rm -rf /content/OID
2 !python3 /content/OIDv4_ToolKit/main.py downloader --classes 'Box' --type_csv all

OIDv4
Downloader

[INFO] | Downloading Box.
[ERROR] | Missing the class-descriptions-boxable.csv file.
[DOWNLOAD] | Do you want to download the missing file? [Y/n] Y
...145%, 0 MB, 43072 KB/s, 0 seconds passed
[DOWNLOAD] | File class-descriptions-boxable.csv downloaded into OID/csv_folder/class-descriptions-boxable.csv.
[ERROR] | Missing the train-annotations-bbox.csv file.
[DOWNLOAD] | Do you want to download the missing file? [Y/n] Y
...100%, 1138 MB, 35650 KB/s, 32 seconds passed
[DOWNLOAD] | File train-annotations-bbox.csv downloaded into OID/csv_folder/train-annotations-bbox.csv.

Box
[INFO] | Downloading all images.
[INFO] | [INFO] Found 2212 online images for train.
[INFO] | Download of 2212 images in train.
100% 2212/2212 [24:51<00:00, 1.48it/s]
[INFO] | Done!
[INFO] | Creating labels for Box of train.
[INFO] | Labels creation completed.
[ERROR] | Missing the validation-annotations-bbox.csv file.
[DOWNLOAD] | Do you want to download the missing file? [Y/n] Y
...100%, 16 MB, 48746 KB/s, 0 seconds passed
[DOWNLOAD] | File validation-annotations-bbox.csv downloaded into OID/csv_folder/validation-annotations-bbox.csv.
```

Рисунок 20 – Пример работы утилиты OIDv4 ToolKit

Произведем загрузку размеченных изображений для домена «Вох»

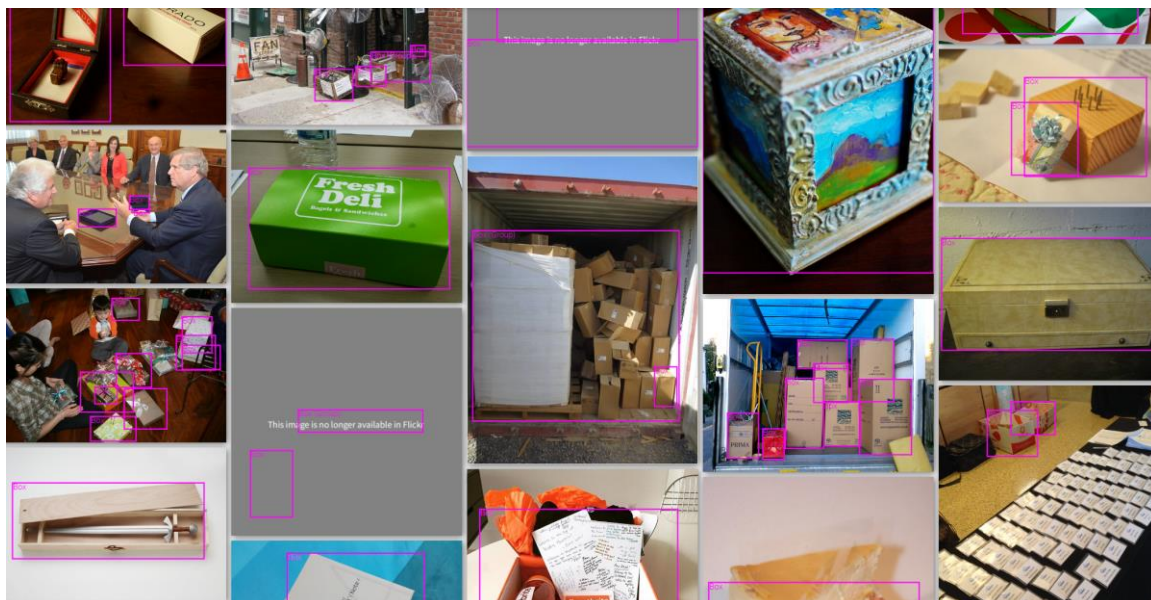


Рисунок 21 – Размеченные изображения в датасете OpenImages

### 2.2.2.3 Преобразование форматов аннотаций OpenImage для дальнейшего использования с Yolo

По результатам экспорта пользовательского набора данных OpenImage у нас имеется несколько тысяч размеченных изображений и аннотаций к ним, но для использования этих данных с YOLOv7 требуется привести аннотацию к совместному с YOLOv7 виду.

Для конвертации аннотаций к изображениям используем сервис Roboflow.

Roboflow — это платформа, предназначенная для работы с компьютерным зрением и обработкой изображений. Она предоставляет набор инструментов и функций, которые помогают разработчикам и исследователям упростить и ускорить процесс создания и развертывания моделей компьютерного зрения.

Для конвертации в нужный формат требуется загрузить датасет в отдельный проект на сайте Roboflow и произвести настройки разметки и преобразования датасета.

После обработки датасета в сервисе Roboflow, набор данных готов для экспорта в требуемый формат и использования с помощью API.



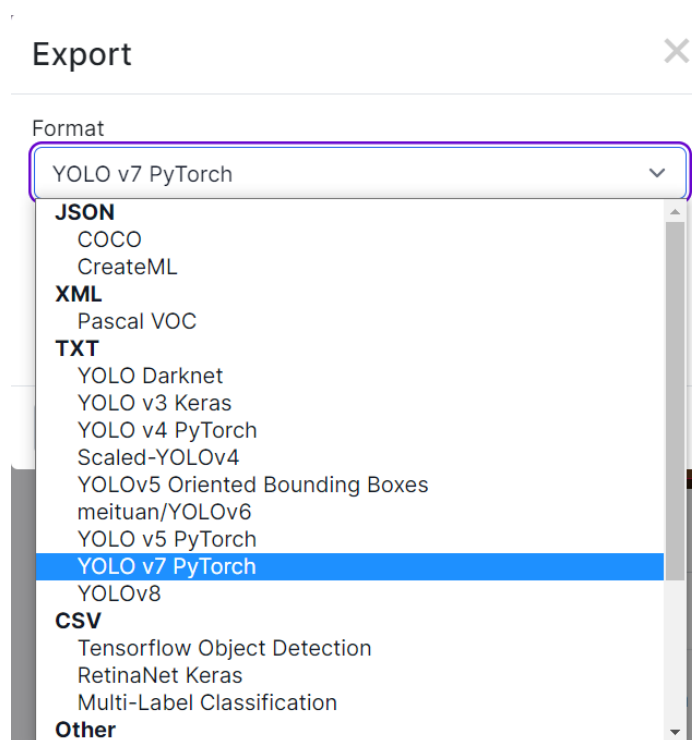


Рисунок 22 – Список доступных для экспорта форматов в сервисе Roboflow

После конвертации данных в формат YOLOv7 формируется файл `data.yaml`. Теперь данные готовы к использованию.

#### 2.2.2.4 Обучение модели YOLO

Для начала произведем детекцию объектов на видео без дополнительного обучения модели, то есть на уже рассчитанных весах по умолчанию. В данном случае будет модель будет детектировать все объекты, для которых рассчитаны стандартные веса `yolov7_training.pt`. В данном случае это объекты набора данных COCO.

COCO (Common Objects in Context) – это один из наиболее широко используемый набор данных в области компьютерного зрения. Он разработан для задач обнаружения объектов, сегментации и классификации в реальных сценах. COCO содержит более 200 тысяч изображений, размеченных с помощью ограничивающих рамок для более чем 80 классов объектов, таких как люди, автомобили, животные, мебель и т. д. Кроме того, набор данных также содержит информацию о сегментации объектов, ключевых точках и классификации.



Произведем тестовую детекцию на все объекты доступные в наборе данных СОСО. В качестве входных данных для детекции будем использовать видеозапись с городской веб-камеры.

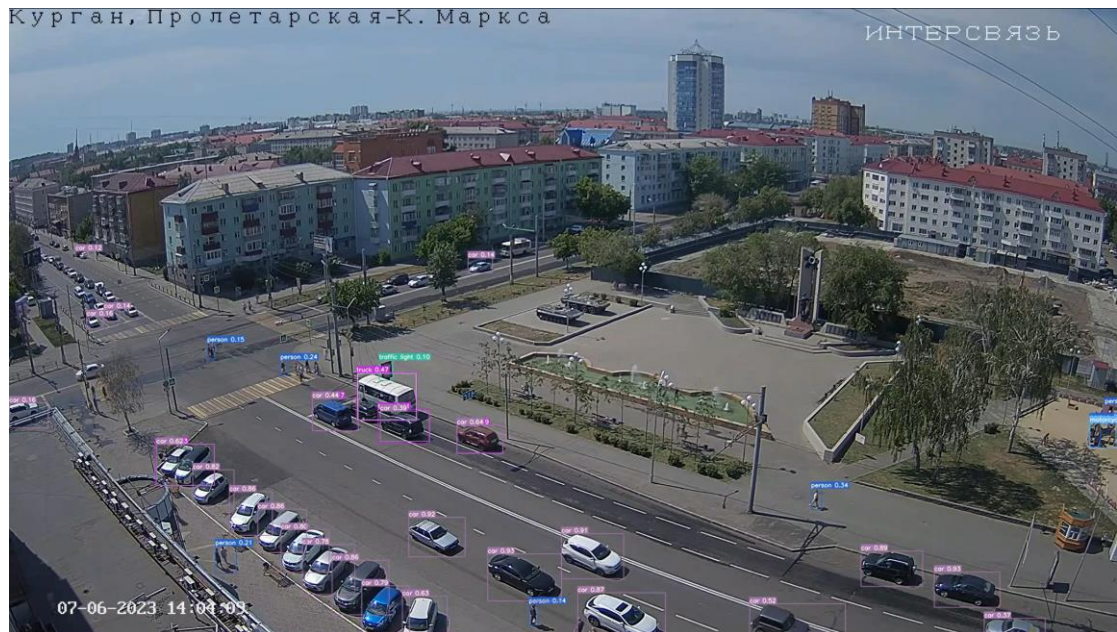


Рисунок 23 – Кадр из видео после детекции на предобученной модели

Видим, что предобученная модель детектирует различные объекты, такие как: пешеходы, машины, светофоры, автобусы, велосипеды.

Теперь попробуем произвести детекцию на более профильном видео в рамках складской тематики, детекция будет производиться на видеозаписи с действующего склада:



Рисунок 24 – Кадр из видео после детекции на предобученной модели

В данном случае модель довольно точно определяет всех людей в помещении, но детектирует всего несколько коробок в класс «Suitcase», что является родственным, но все-таки ошибочным классом.

Теперь произведем детекцию на более качественной видеозаписи и с более удачного ракурса.



Рисунок 25 – Ошибочная детекция коробок на предобученной модели

В данном случае модель принимает открытые коробки за класс «chair» или не детектирует вовсе.

На данном изображении коробка игнорируется и никак не детектируется, зато определяются объекты других классов, таких как «Suitcase» и «Sink».



Рисунок 26 – пример детекции на предобученной модели

Обучим модель YOLOv7 на ранее подготовленных данных OpenImage. В качестве площадки для обучения будем использовать интерактивную среду Google Colaboratory, в качестве аппаратного ускорителя выбираем GPU (графический процессор T4). Производим обучение длительностью 25 эпох.

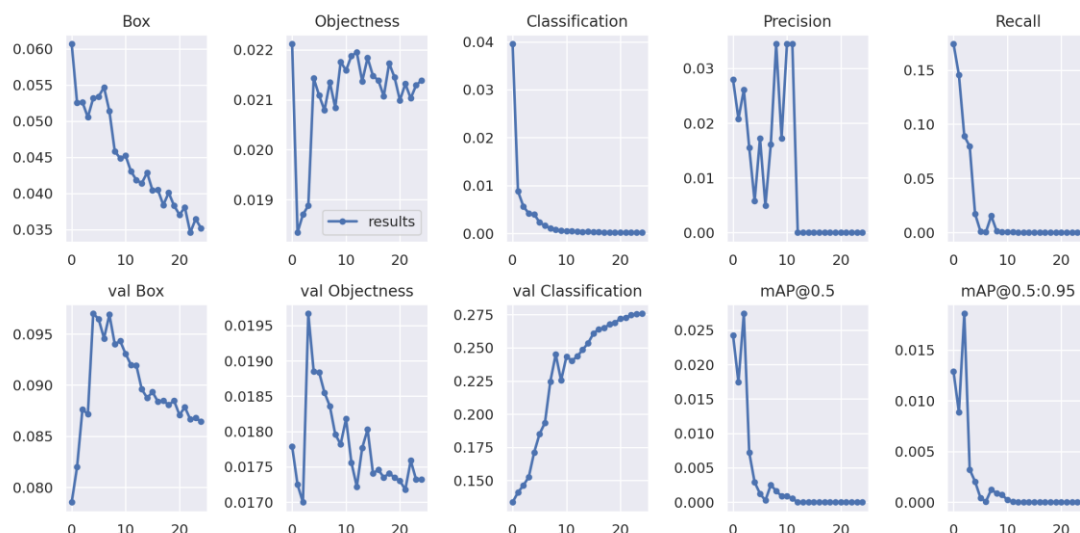


Рисунок 27 – метрики обучения на пользовательском наборе данных

Далее производим детекцию объекта «Вох» по видеозаписи. По фрагменту видеоряда фиксируем что в отличии от рисунка 26 на рисунке 28 отсутствуют bbox классов «Suitcase» и «Sink», но зато определяет новый пользовательский класс «Вох»(Упаковка с товаром) и детектирует его на протяжении всего видеоряда.



Рисунок 28 – Детекция объекта «Box» с использованием дообученной модели

Для того чтобы улучшить показатели метрик предлагаются следующие улучшения:

- исключить из датасета данные неудовлетворяющего качества;
- исключить ошибки, неточности или неправильную разметку объектов;
- настроить гиперпараметры модели для обучения;
- оптимизировать время обучения, исключая переобучение.

### 2.2.3 Использование модели MediaPipe для детекция 3d-объектов

MediaPipe – это фреймворк с открытым исходным кодом от Google, который предоставляет инструменты для создания видеоаналитических приложений. Он позволяет обнаруживать и отслеживать объекты, анализировать движение, жесты и мимику лица, а также выполнять другие операции обработки видео и аудио.

MediaPipe предоставляет готовые модели и компоненты для детекции 3D-объектов, такие как Objectron, который обнаруживает трехмерные объекты в реальном времени. Кроме того, с помощью инструментов и библиотек MediaPipe можно создавать собственные модели и компоненты, адаптированные под конкретные задачи и требования.



MediaPipe Objectron – это компонент фреймворка, предназначенный для обнаружения и оценки 3D-объектов в реальном времени. Он использует нейронные сети и компьютерное зрение для анализа видеопотока и определения позиции и размеров объектов в сцене.

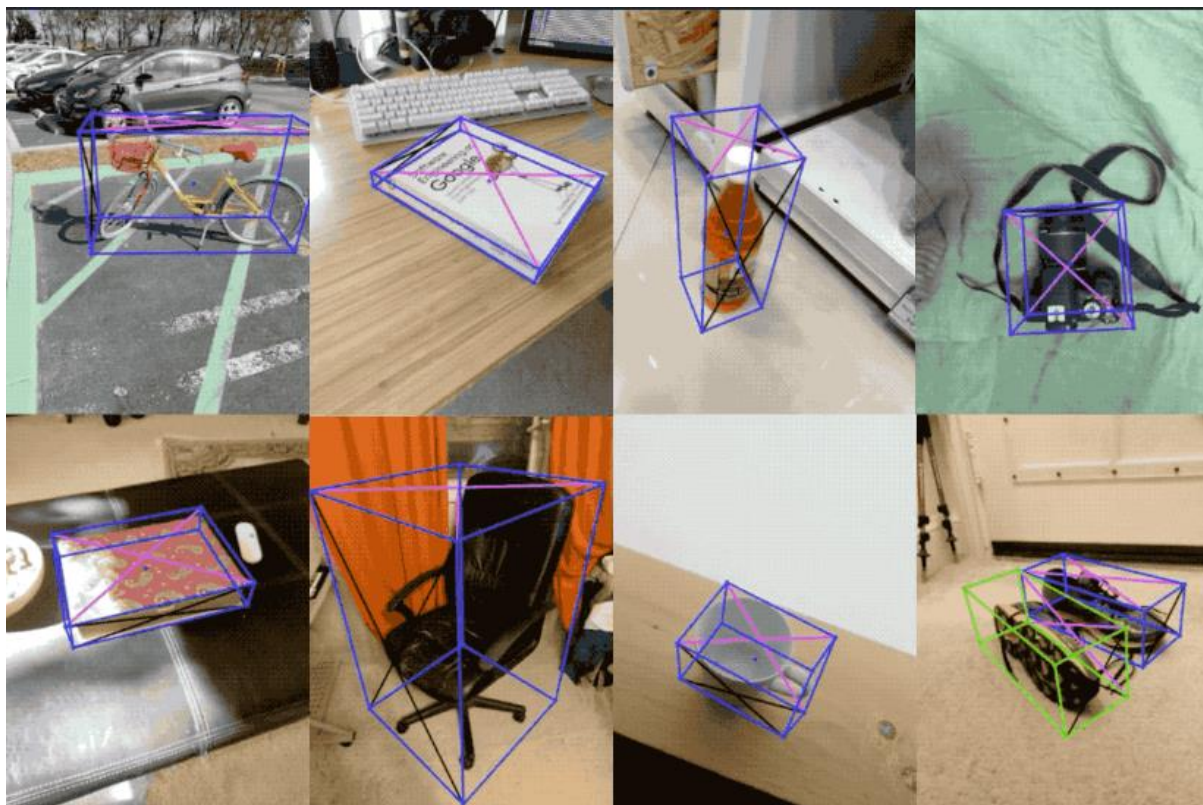


Рисунок 29 – пример детекции 3d-объектов модели MediaPipe Objectron

Основные возможности и характеристики MediaPipe Objectron:

- обнаружение 3D-объектов: Objectron позволяет обнаруживать различные типы 3D-объектов, такие как сумки, обувь, стулья, бутылки и другие повседневные предметы. Он использует глубокие нейронные сети для обнаружения и классификации объектов на основе визуальных данных;
- оценка 3D-позиции и размеров: Objectron оценивает 3D-позицию и размеры обнаруженных объектов в сцене. Это позволяет получить информацию о положении объектов в пространстве и их геометрических характеристиках;
- работа в режиме реального времени: Objectron спроектирован для работы в реальном времени, что означает, что он может обрабатывать видеопотоки в реальном времени, что означает, что он может обрабатывать видеопотоки

токи в режиме практически мгновенного времени отклика. Это позволяет использовать Objectron в приложениях, где требуется непрерывный анализ и отслеживание 3D-объектов в реальном времени.

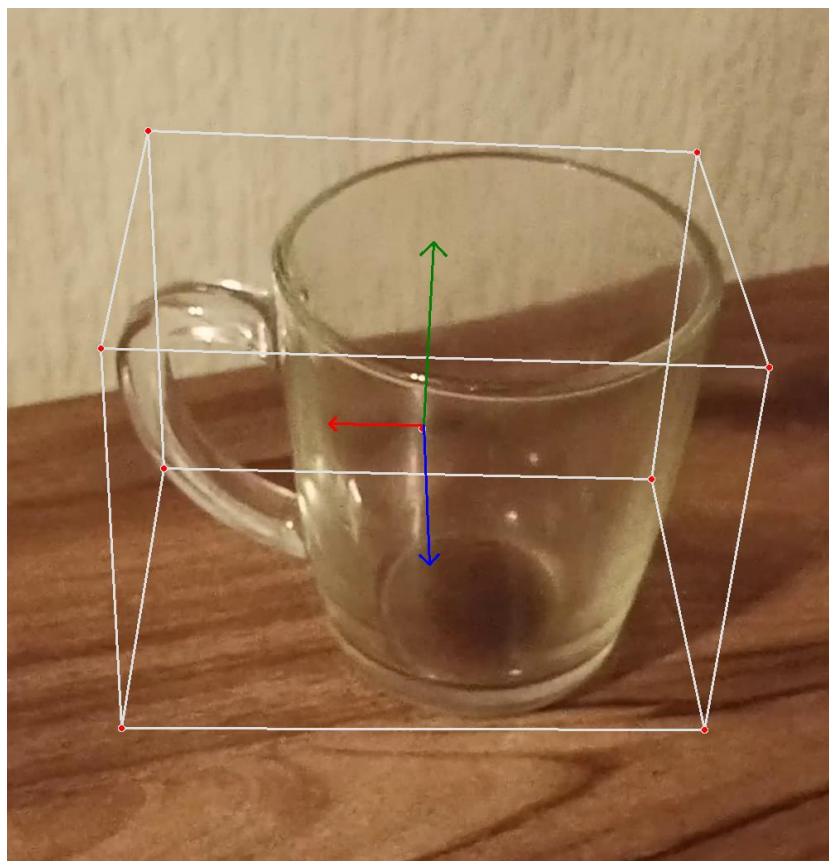


Рисунок 30 – Пользовательская детекция 3d-объекта класса «Cup»

### 2.3 Вывод к главе

В рамках данной главы был проведен анализ применения глубокого обучения и нейронных сетей в области складской логистики. Были сформулированы различные варианты применения глубокого обучения и нейронных сетей в этой области, а также предложено переформулирование данных вариантов к классическим задачам компьютерного зрения.

Была представлена методология исследования, включающая модель компьютерного зрения YOLO, предназначенная для детектирования объектов. В качестве тестового набора данных был рассмотрен датасет OpenImages, который включает в себя класс «Box» с размеченными изображениями различных упаковок и коробок. Рассмотрены важные аспекты подготовки входных данных для

модели YOLO, включая преобразование форматов аннотаций OpenImage и обучение модели на пользовательском наборе данных.

Для дальнейшего исследования предлагается рассмотреть область сегментации объектов, которая позволит улучшить точность получаемых показателей. Также можно рассмотреть возможность детекции 3d-объектов на основе на основе двумерных изображений.

В заключении можно отметить, что глубокое обучение и нейронные сети имеют значительный потенциал в области складской логистики. Они способны справиться с задачами компьютерного зрения, такими как детекция и классификация объектов на складе, оптимизация маршрутов и управление инвентарем. Применение моделей, таких как YOLO и MediaPipe, демонстрирует эффективность и точность в решении этих задач.

Однако, необходимо отметить, что успешная реализация данных методов требует качественных и разнообразных наборов данных для обучения моделей, а также оптимизации параметров и архитектуры сетей под конкретные задачи. Дальнейшие исследования и разработки в этой области могут привести к ещё более точным и эффективным решениям в складской логистике, что способствует повышению производительности и сокращению затрат в этой отрасли.



### **3 Разработка системы компьютерного зрения для детекции и сегментации товаров на сортировочной ленте**

Целью данной главы является исследование методов повышения точности детектирования объектов на изображениях и интеграция техник сегментации в контексте складской логистики. Компьютерное зрение играет ключевую роль в автоматизации и оптимизации процессов. Складская логистика сталкивается с многочисленными вызовами, включая необходимость быстрого и точного распознавания и сортировки товаров, требующих инновационных подходов в области компьютерного зрения. Данная глава ориентирована на адаптацию алгоритмов глубокого обучения для улучшения точности детекции и сегментации объектов в условиях, характерных для складских помещений, что включает в себя как улучшение предварительной обработки изображений, так и оптимизацию гиперпараметров и функций потерь. Особое внимание уделяется возможности применения синтетических и дополненных данных для увеличения эффективности обучения моделей.

В рамках методологического подхода рассматривается решение задачи детекции с использованием современной модели YOLOv8, которая демонстрирует впечатляющие результаты в обнаружении объектов в различных условиях. Помимо применения этой модели, в главе освещается процесс дообучения пользовательской модели детекции для специфических нужд складской логистики, а также методы приведения пользовательских датасетов к единой аннотации, что обеспечивает согласованность и повышает качество обучения.

Далее, исследование переходит к задаче сегментации. Здесь ключевым моментом является использование модели SAM для сегментации объектов. SAM, или Segment Anything Model, представляет собой передовой подход в выделении и сегментации интересующих объектов, что особенно важно для точного и эффективного определения габаритных характеристик. В рамках главы также рассматривается создание набора данных для сегментации на основе обнаруженных моделью YOLOv8 объектов.

### **3.1 Методы повышения точности детектирования объектов на изображениях**

#### **3.1.1 Улучшение предварительной обработки изображений**

Нормализация и стандартизация данных. Этот шаг включает в себя масштабирование пиксельных значений изображений для улучшения стабильности и скорости обучения нейронных сетей. Нормализация часто включает преобразование диапазона пиксельных значений изображения к  $[0, 1]$  или  $[-1, 1]$  путем деления на 255 (максимальное значение пикселя). Стандартизация включает в себя центрирование распределения пикселей изображения вокруг нуля и нормализацию его дисперсии.

Аугментация данных используется для увеличения разнообразия обучающих данных и предотвращения переобучения, используются техники аугментации изображений. Это включает в себя:

- геометрические преобразования, такие как вращение, масштабирование, и сдвиги;
- изменение цветового пространства, включая коррекцию яркости, контрастности, насыщенности, и оттенка;
- применение случайных обрезаков и перспективных трансформаций;
- использование зеркального отображения и случайных искажений;
- введение шума, такого как гауссовский или «соль-перец».

На рисунке 31 изображен пример преобразованного входного изображения.

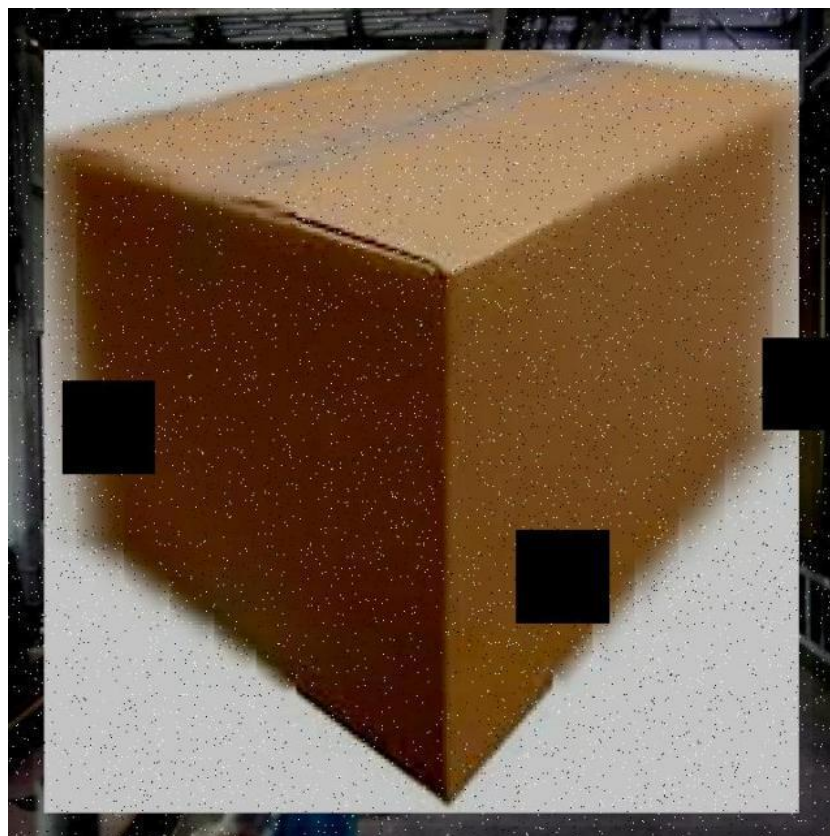


Рисунок 31 – Пример аугментации данных

Фильтрация и улучшение качества изображения: Применение различных фильтров для улучшения визуальных особенностей изображений и уменьшения шума:

- фильтры размытия (например, Гауссово размытие) для сглаживания шумов;
- фильтры улучшения резкости для выделения краев и деталей изображения;
- морфологические операции (эрозия, дилатация) для улучшения структуры объектов.

### 3.1.2 Оптимизация гиперпараметров

Оптимизация гиперпараметров – это эффективный подбор и настройка гиперпараметров. Значения гиперпараметров существенно влияют на производительность модели.

Важные гиперпараметры включают:

- скорость обучения. Определение начальной скорости обучения и стратегии ее изменения в процессе обучения (например, с помощью планировщиков скорости обучения или адаптивных оптимизаторов);
- размер пакета. Выбор подходящего размера пакета, который сбалансирует скорость обучения и стабильность;
- количество эпох. Определение оптимального числа эпох для обучения модели без переобучения.

Модификация параметров применения модели:

- определение порогов уверенности детекции или сегментации;
- количество слоев для дополнительных предсказаний маски на обрезанных изображениях;
- определяет соотношения перекрытия между сегментированными элементами изображения;
- пользовательские тензоры и координатные сетки для выбора точек;
- размер пакета для одновременной обработки количества точек;
- порог IoU для NMS (non-maximum suppression) для удаления дублирующих масок между частями;
- разнообразные подсказки (bbox, маски, точки).

### 3.1.3 Применение синтетических и дополненных данных

Генерация синтетических данных используется для повышения обучающей способности модели, генерация синтетических данных может быть использована для дополнения обучающего набора:

- использование GAN (Generative adversarial network,). Генеративно-сопоставительные сети могут создавать реалистичные изображения для увеличения разнообразия данных;
- симуляции. В некоторых случаях, особенно в робототехнике, симуляции могут быть использованы для генерации данных обучения.

Дополнение реальных данных синтетическими примерами и интеграция синтетических данных с реальными изображениями для создания более обширного и разнообразного набора данных:

- разработка сценариев для синтетических данных. Создание сценариев, которые эффективно дополняют реальные данные и улучшают обучение в специфических случаях;
- смешивание реальных и синтетических данных. Использование различных стратегий для смешивания синтетических и реальных данных, чтобы улучшить обучение и обеспечить реалистичность обучающего набора.

Симуляции и 3D моделирование цифровых двойников может повысить эффективность обучения алгоритмов компьютерного зрения, особенно в условиях, где сбор реальных данных затруднителен, дорог или опасен. Симуляции позволяют создавать контролируемые условия, такие как различные погодные условия и освещение, что особенно полезно в робототехнике и промышленности. Симуляции также могут облегчить процесс сбора аннотации, например, ограничивающих рамок и пиксельных масок, что значительно экономит время и усилия по сравнению с ручной аннотацией. Кроме того, симуляции позволяют легко манипулировать объектами и их свойствами, создавая обширные и разнообразные наборы данных, которые масштабируемы и позволяют генерировать большие объемы данных без значительных дополнительных затрат.

Кроме того, они могут использоваться для анализа и оптимизации производительности систем, позволяя проводить эксперименты и тестирования в безопасной и контролируемой среде, а также могут быть использованы для тестирования алгоритмов машинного обучения перед их применением в реальном мире.

Таким образом, сочетание симуляций и 3D моделирования цифровых двойников предоставляет мощные инструменты для разработки, тестирования и улучшения алгоритмов компьютерного зрения, обеспечивая более широкий

спектр обучающих данных и возможности для тестирования в разнообразных и контролируемых средах.

### 3.2 Архитектурный анализ и проектирование системы

Данный раздел посвящен проектированию диаграммы IDEF0, диаграммы классов и анализу архитектурных решений для интеграции системы SortifyScan в существующие системы.

#### 3.2.1 Диаграмма IDEF0

Данный раздел посвящен диаграмме IDEF0.

На рисунке 32 изображена диаграмма главного процесса.

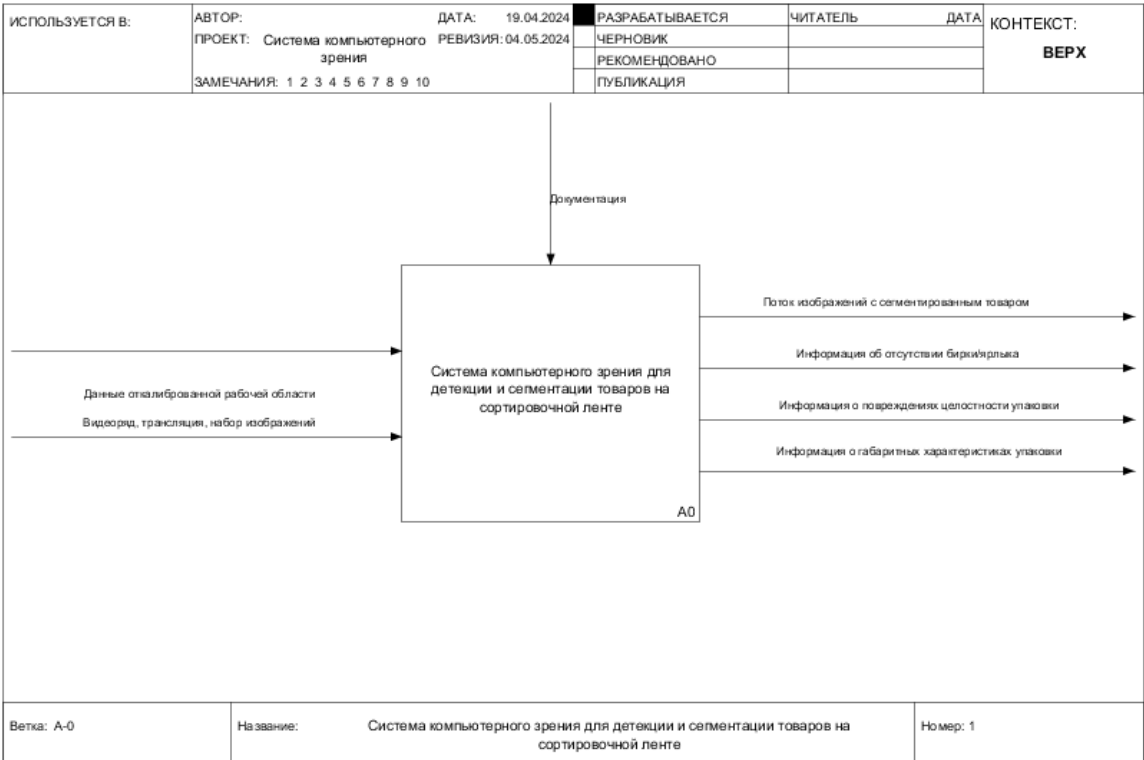


Рисунок 32 – Диаграмма главного процесса

Название процесса: Система компьютерного зрения для детекции и сегментации товаров на сортировочной ленте.

На вход поступают:

- данные откалиброванной области;
- видеоряд, трансляция, набор изображений.

Результат процесса:

- поток изображений с сегментированным товаром;
- информация об отсутствии бирки/ярлыка;

- информация о габаритных характеристиках;
- информация о наличии повреждений.

На рисунке 33 изображена декомпозиция главного процесса.

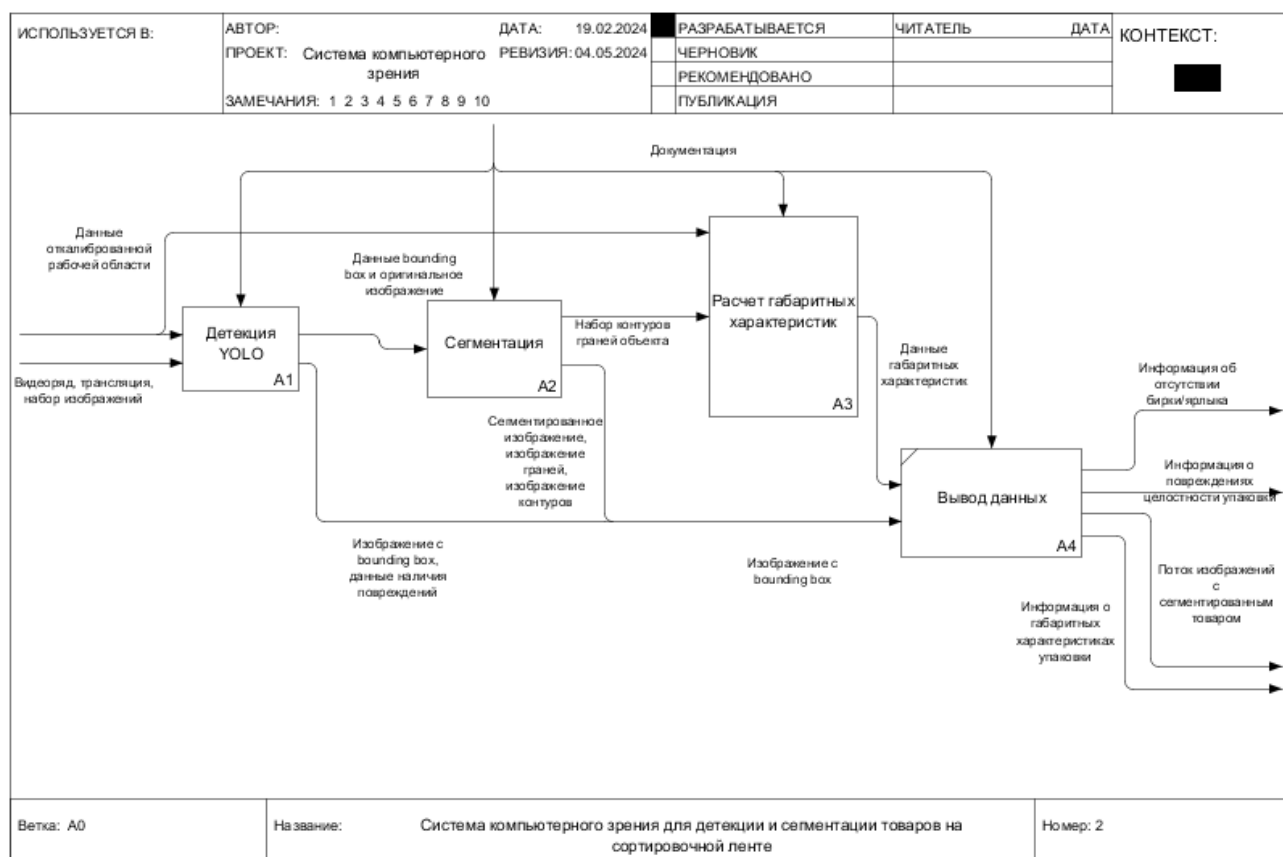


Рисунок 33 – Декомпозиция главного процесса

#### 1) Детекция YOLO:

- этот процесс отвечает за первичную детекцию объектов с использованием модели YOLO.
- на вход этого процесса поступают данные откалиброванной области и видеоряд, трансляция, набор изображений
- результатом работы этого процесса является изображение с bounding box и информация об отсутствии бирки/ярлыка.

#### 2) Сегментация:

- в этом процессе изображения объектов сортировочной ленты сегментируются для выделения их контуров.
- для выполнения сегментации используются данные откалиброванной области.

с. результатом этого процесса является сегментированное изображение.

### 3) Расчет габаритных характеристик:

- а. этот процесс выполняет расчет габаритных характеристик обнаруженных объектов, таких как размеры и форма.
- б. для расчета используются данные откалиброванной области.
- с. результатом работы этого процесса является информация о габаритных характеристиках упаковки.

### 4) Вывод данных:

- а. этот процесс представляет собой финальный этап системы, где готовые данные выводятся для последующей обработки или анализа.
- б. результатами работы этого процесса являются поток изображений с сегментированным товаром, информация об отсутствии бирки/ярлыка, информация о повреждениях целостности упаковки и информация о габаритных характеристиках упаковки.

На рисунке 34 изображена декомпозиция процесса «Детекция YOLO».

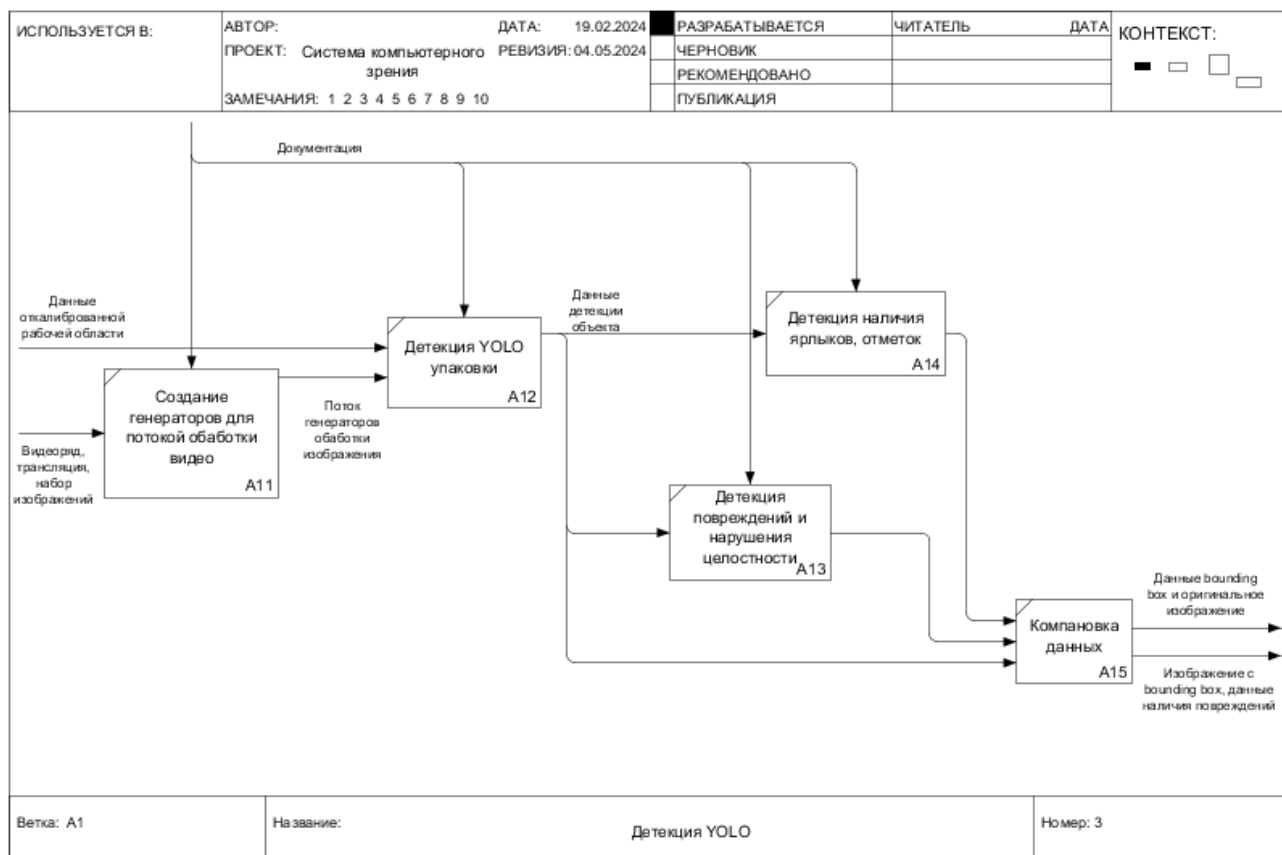


Рисунок 34 – Декомпозиция процесса «Детекция YOLO»



## Процесс «Детекция YOLO»:

- 1) Создание генераторов для потоковой обработки видео:
  - a. этот блок отвечает за создание генераторов, необходимых для обработки видеопотока;
  - b. вход: видеоряд, трансляция, набор изображений;
  - c. выход: поток генераторов обработки изображения.
- 2) Детекция YOLO упаковки:
  - a. в этом блоке осуществляется детекция объектов упаковки с использованием алгоритма yolo;
  - b. вход: данные откалиброванной рабочей области;
  - c. выход: данные детекции объекта.
- 3) Детекция повреждений и нарушения целостности:
  - a. в этом блоке происходит детекция повреждений и нарушений целостности упаковки;
  - b. вход: данные детекции объекта;
  - c. выход: данные детекции объекта.
- 4) Детекция наличия ярлыков, отметок:
  - a. этот блок отвечает за детекцию наличия ярлыков и других отметок на упаковке;
  - b. вход: данные детекции объекта;
  - c. выход: данные детекции объекта.
- 5) Компановка данных:
  - a. этот блок представляет собой финальный этап, где данные от всех предыдущих блоков komponуются для дальнейшей обработки или анализа;
  - b. вход: данные детекции объекта и данные детекции повреждений;
  - c. выход: данные bounding box и оригинальное изображение.

На рисунке 35 изображена декомпозиция процесса «Сегментация».

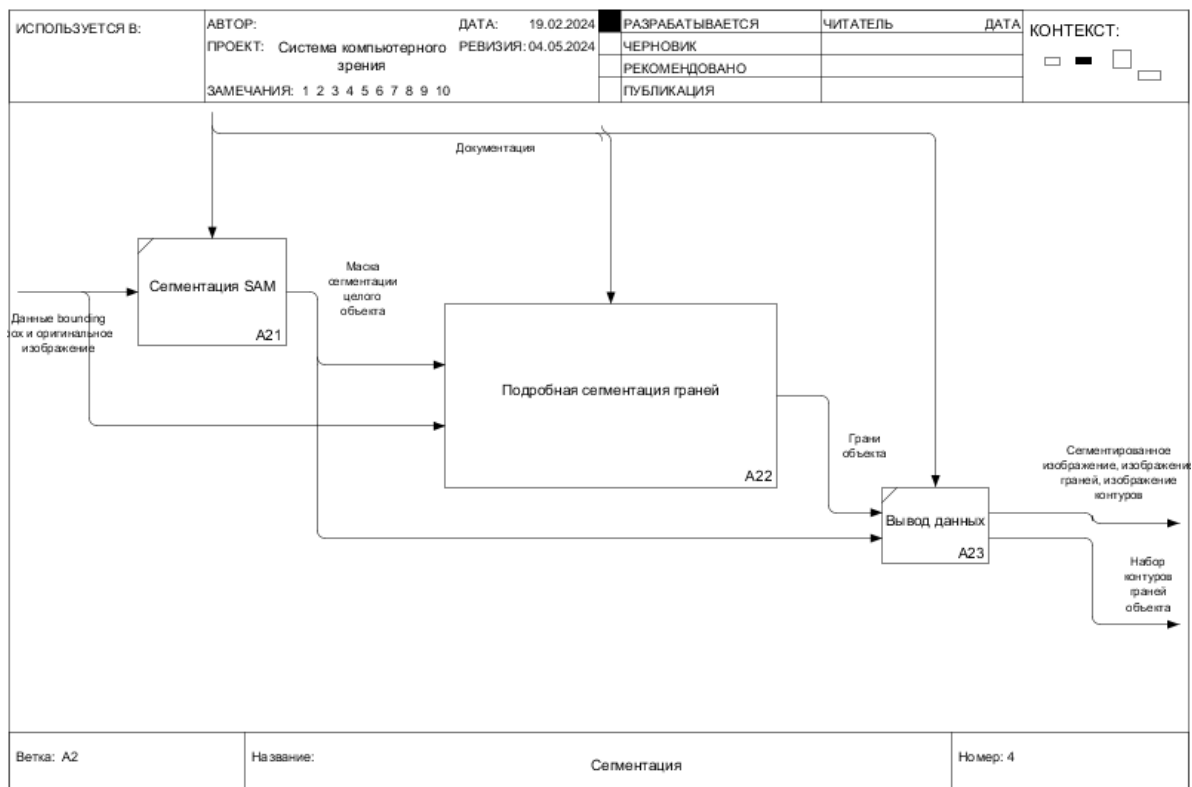


Рисунок 35 – Декомпозиция процесса «Сегментация»

Декомпозиция процесса «Сегментация»:

1) Сегментация SAM:

- а. этот блок отвечает за применение алгоритма сегментации SAM для выделения объектов на изображении;
- б. вход: данные bounding box и оригинальное изображение;
- с. выход: маска сегментации целого объекта.

2) Подробная сегментация граней:

- а. здесь осуществляется подробная сегментация граней объекта для получения дополнительной информации о его структуре;
- б. вход: маска сегментации целого объекта;
- с. выход: грани объекта.

3) Вывод данных:

- а. этот блок представляет собой завершающий этап, где данные сегментации компонуются и выводятся;
- б. вход: маска сегментации целого объекта и грани объекта;
- с. выход: набор контуров граней объекта.

На рисунке 36 изображена декомпозиция процесса «Подробная сегментация граней».

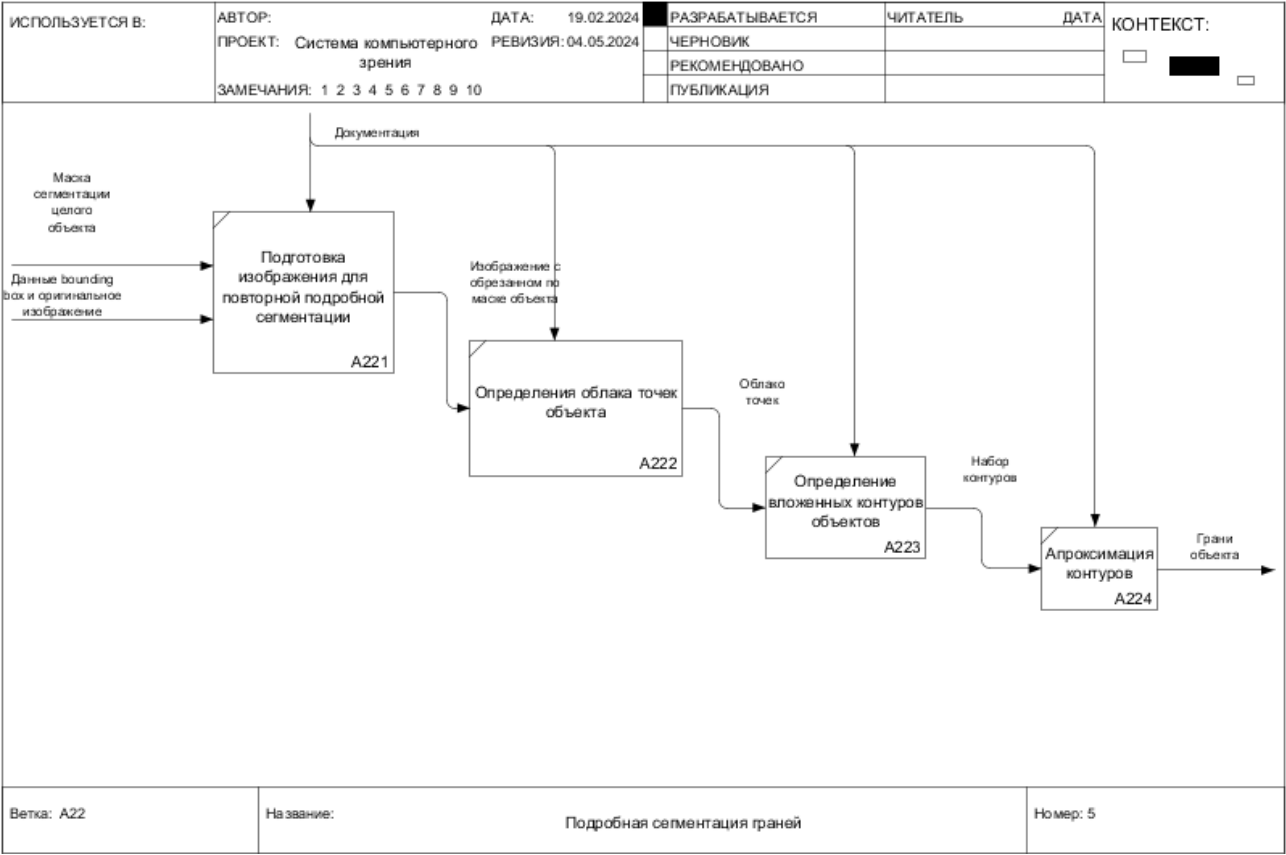


Рисунок 36 – Декомпозиция процесса «Подробная сегментация граней»  
Процесс «Подробная сегментация граней»:

- 1) Подготовка изображения для повторной подробной сегментации:
  - а. этот блок выполняет подготовку изображения для повторной подробной сегментации объекта;
  - б. вход: данные bounding box и оригинальное изображение, а также маска сегментации целого объекта;
  - с. выход: изображение, обрезанное по маске объекта.
- 2) Определения облака точек объекта:
  - а. здесь происходит определение облака точек объекта на основе обрезанного изображения;
  - б. вход: изображение, обрезанное по маске объекта;
  - с. выход: Облако точек объекта.
- 3) Определение вложенных контуров объектов:

- а. в этом блоке происходит определение вложенных контуров объектов на основе облака точек;
- б. вход: облако точек объекта;
- с. выход: набор контуров объекта.

4) Аппроксимация контуров:

- а. этот блок выполняет аппроксимацию контуров объекта для получения граней;
- б. вход: набор контуров объекта;
- с. выход: грани объекта.

На рисунке 37 изображена декомпозиция процесса «Расчет габаритных характеристик».

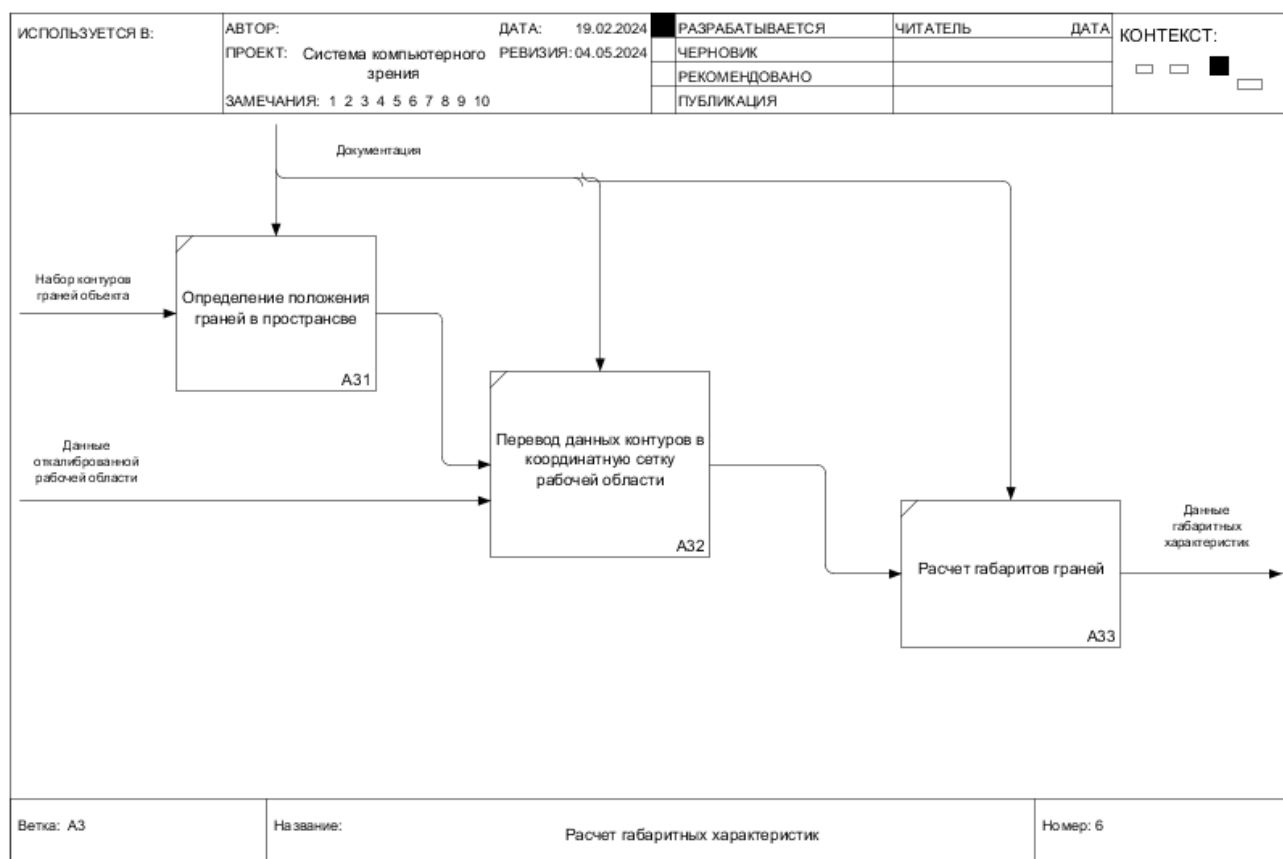


Рисунок 37 – Декомпозиция процесса «Расчет габаритных характеристик»

Процесс «Расчет габаритных характеристик»:

1) Определение положения граней в пространстве:

- а. этот блок определяет положение граней объекта в пространстве;

- b. вход: набор контуров граней объекта;
- c. выход: Данные габаритных характеристик.

2) Перевод данных контуров в координатную сетку рабочей области:

- a. здесь происходит перевод данных контуров в координатную сетку рабочей области;
- b. вход: набор контуров граней объекта;
- c. выход: Данные габаритных характеристик.

3) Расчет габаритов граней:

- a. в этом блоке производится расчет габаритов граней на основе данных, полученных от предыдущих блоков;
- b. вход: данные габаритных характеристик;
- c. выход: данные габаритных характеристик.

### 3.2.2 Диаграмма развертывания

Диаграмма развертывания показывает физическое расположение и взаимодействие различных компонентов системы.

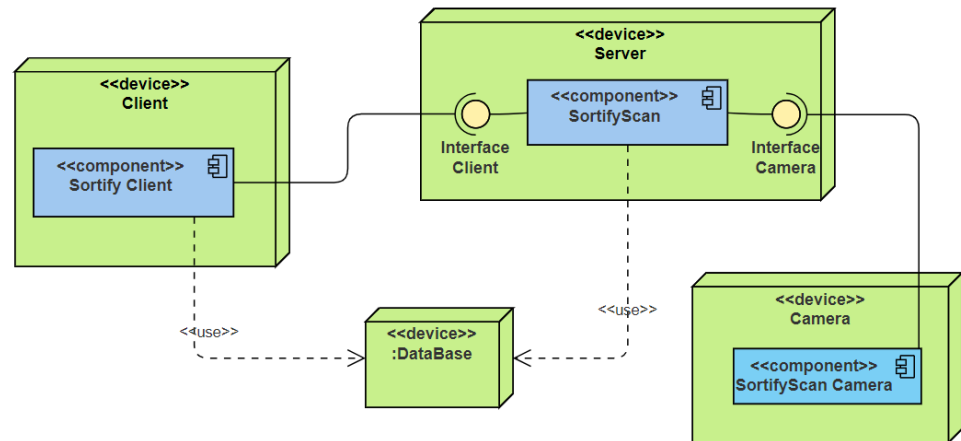


Рисунок 38 – Дигарамма развертывания

Компоненты диаграммы развертывания:

- Клиентская часть (Client). Пользователь взаимодействует с системой через интерфейс, который отправляет запросы на сервер и отображает результаты детекции и сегментации.
- Камера (Camera). Устройство, которое захватывает изображения товаров на сортировочной ленте и передает их на сервер.
- Сервер (Server). Основной компонент, где происходит обработка изображений. Здесь выполняются детекция и сегментация объектов, расчет габаритных характеристик и детекция повреждений. После обработки данные отправляются в базу данных.
- База данных (Database). Сохраняет результаты обработки, включая информацию о габаритных характеристиках, сегментации и детекции повреждений, для дальнейшего анализа и использования.

### 3.2.3 Диаграмма классов

На рисунке 39 представлена диаграмма классов.

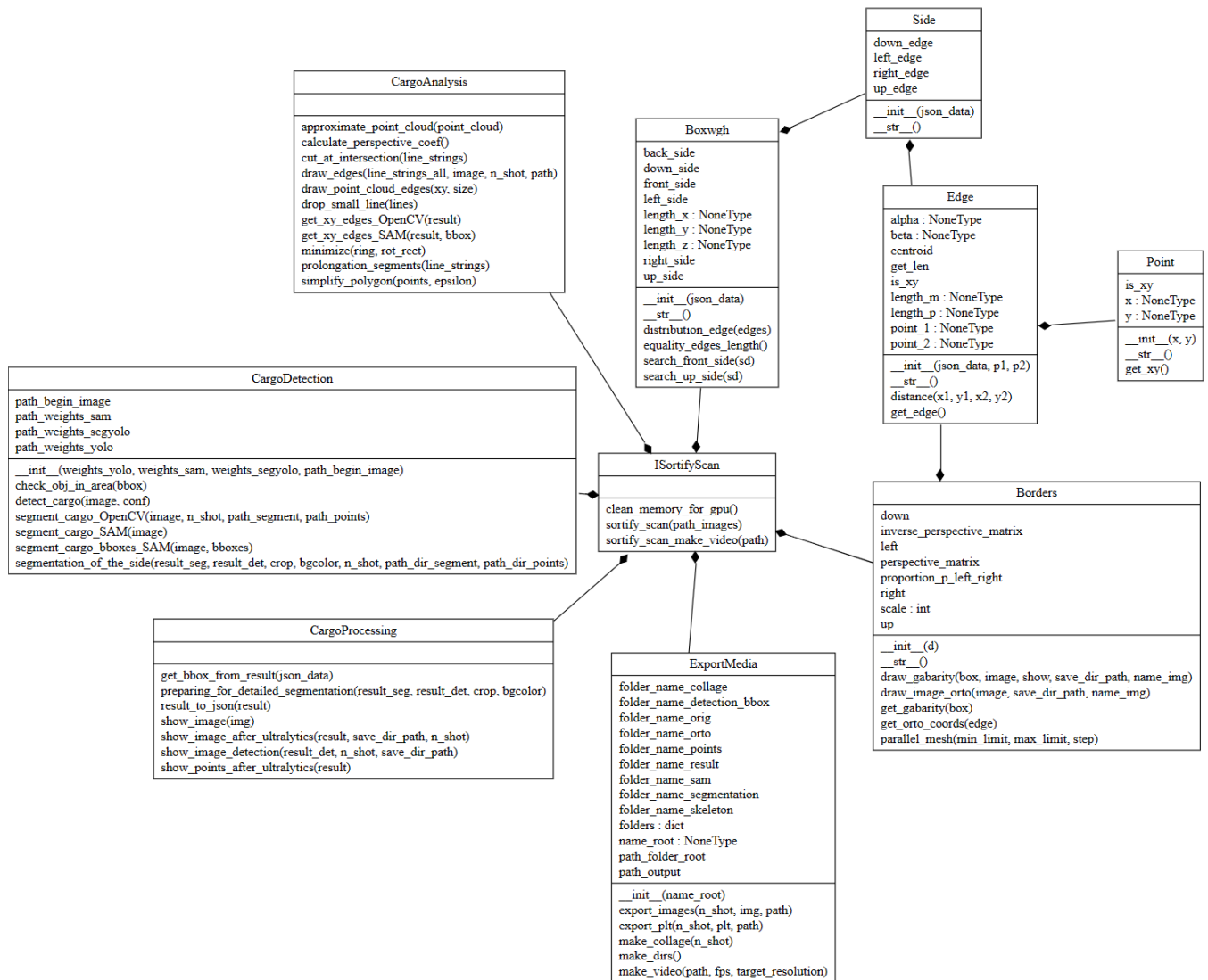


Рисунок 39 – Диаграмма классов SortifyScan

Система SortifyScan состоит из классов:

- ISortifyScan. интерфейс для начала работы с системой. Содержит методы для подготовки данных для GPU, запуска сортировки сканирования и создания видео сканирования.
- CargoAnalysis. Этот класс представляет анализ груза. Он содержит методы для анализа облака точек, определения перспективного коэффициента, аппроксимации, упрощения полигональных точек и т. д.

- `CargoDetection`. Класс для обнаружения груза. Содержит методы для обнаружения груза на изображениях, сегментации груза и проверки объекта в области.
- `CargoProcessing`. Этот класс представляет обработку груза. Он содержит методы для подготовки детализированной сегментации, отображения результатов и преобразования результатов в формат JSON.
- `ExportMedia`. Класс для экспорта медиа. Он содержит методы для экспорта изображений, создания видео и кадров.
- `Borders`. Этот класс представляет собой границы. Он содержит атрибуты и методы для работы с границами, такие как матрицы, масштабирование и методы для отрисовки.
- `Boxwgh`. Класс, представляющий габаритный параллелепипед. Он содержит атрибуты и методы для работы с параллелепипедом, такие как определение сторон, поиск граничных сторон и методы распределения краев.
- `Side`. Класс, представляющий сторону. Он содержит атрибуты и методы для работы с гранями.
- `Edge`. Класс, представляющий край. Он содержит атрибуты и методы для работы с краями, такие как расстояние между точками и получение координат ребра.
- `Point`. Этот класс представляет точку. Он содержит атрибуты и методы для работы с точками, такие как получение координат.

Необходимые для успешной работы системы, библиотеки Python:

- `Json`. Для работы с данными в формате JSON;
- `Ultralytics`. Пакет, предоставляющий функциональность для обучения и оценки моделей глубокого обучения;
- `PIL`. Библиотека Python Imaging Library, используемая для работы с изображениями;
- `Matplotlib`. Библиотека для создания графиков и визуализации данных;



- Pathlib. Модуль для манипуляции путями к файлам и папкам;
- OpenCV, библиотека компьютерного зрения и обработки изображений;
- Numpy. Библиотека для работы с многомерными массивами и математическими операциями над ними;
- Shapely. Библиотека для работы с геометрическими объектами, такими как точки, линии и полигоны;
- Torch. Фреймворк глубокого обучения PyTorch;
- Torchvision. Пакет, содержащий наборы данных, архитектуры моделей и трансформации для компьютерного зрения PyTorch.

### 3.2.4 Архитектура возможной интеграции в существующие системы

На рисунке 40 представлена возможная архитектура системы для интеграции в другие системы.

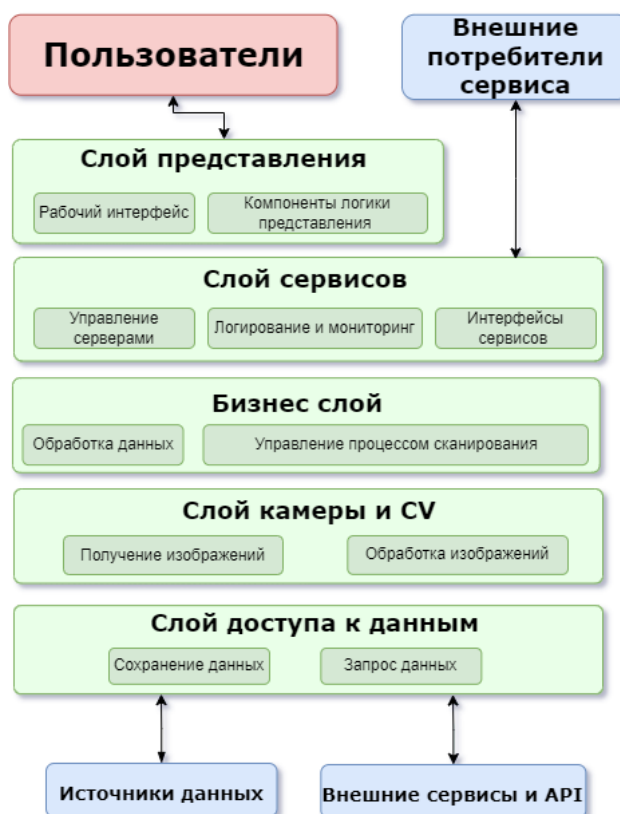


Рисунок 40 – Послойная архитектуры системы

Уровень представления обеспечит пользовательский интерфейс для операторов и работников склада.

Уровень бизнес-логики будет отвечать за обработку данных о товарах и мониторинг процесса сканирования.

Уровень доступа к данным будет взаимодействовать с общей базой данных для сохранения и извлечения информации о товарах.

Уровень сервисов обеспечивает логирование и управление вычислительными системами.

Уровень обработки данных представляет из себя систему детекции и сегментации.

На рисунке 41 представлена возможная общая схема взаимодействия систем.

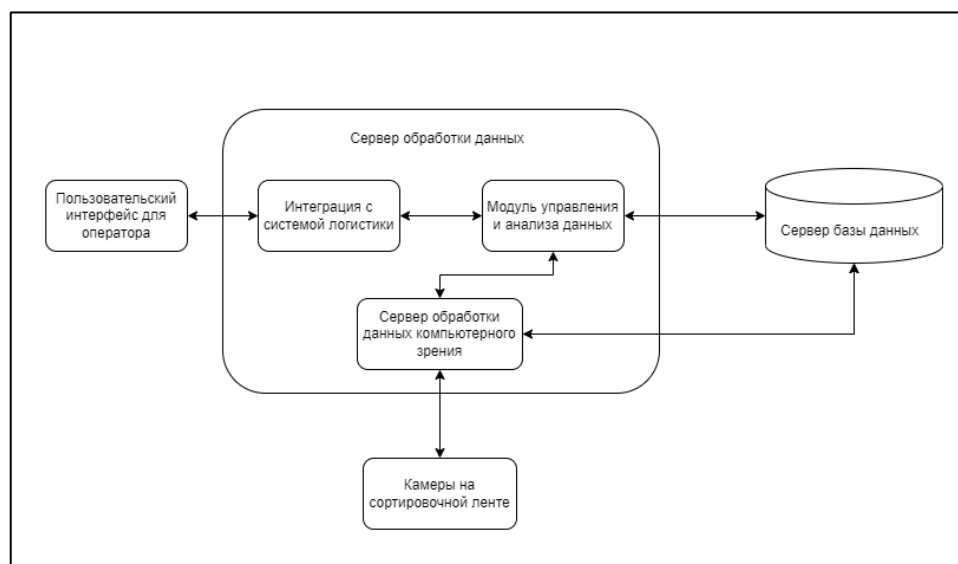


Рисунок 41 – Системная архитектура

### 3.3 Разработка модуля компьютерного зрения для решения задач складской логистики

#### 3.3.1 Решение задачи детекции

В складской логистики задача детекции играет критически важную роль, так как она лежит в основе автоматизации процессов учета и управления запасами. Автоматическое распознавание товаров и их характеристик через компьютерное зрение позволяет значительно ускорить и оптимизировать логистические процессы.

### 3.3.1.1 Применении YOLOv8 для задач обнаружения

YOLOv8, будучи одной из последних итераций в серии моделей YOLO (You Only Look Once), представляет собой мощный инструмент для детекции объектов в реальном времени. Благодаря своей способности быстро и точно локализовать объекты на изображениях, YOLOv8 идеально подходит для применения в складской логистике. Она может использоваться для идентификации товаров, их местоположения на полках, при движении по конвейерной ленте, а также для считывания штрих-кодов или других идентификаторов. Важно отметить, что YOLOv8 обладает высокой точностью и скоростью, что критически важно для временно чувствительных операций функционирования фулфилмента.

В таблице 2 представлены основные показатели доступных в YOLOv8 моделей

Таблица 2 – Показатели производительности YOLOv8.

Модель	Размер	mAPval 50-95	Скорость CPU ONNX(мс)	СкоростьA100 TensorRT(мс)	params(M)	FLOPs(B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

### 3.3.1.2 Дообучение пользовательской модели детекции

Для достижения максимальной эффективности в специфических условиях, требуется дообучение модели детекции. Дообучение (fine-tuning) YOLOv8 на специализированных данных склада позволяет адаптировать модель к конкретным условиям и особенностям работы, таким как особенности освещения, расположения товаров и их внешний вид. Это улучшает точность детекции и снижает количество ложных срабатываний, что особенно важно для автоматизированных систем управления складом.

В YOLOv8 существуют предварительно обученные модели Detect. Модели Detect были предварительно обучены на наборе данных COCO.

Набор данных COCO (Common Objects in Context) — это масштабный набор данных по обнаружению, сегментации объектов и созданию надписей. Он предназначен для стимулирования исследований самых разных категорий объектов и обычно используется для бенчмаркинга моделей компьютерного зрения.

Для детекции кастомных объектов требуется дообучить модель. Для дообучения будем использовать подходящий датасет из открытого источника Roboflow, для использования требуется экспортировать датасет в формате YOLOv8.

Моделью старта возьмем версию весов YOLOv8l.pt.

В результате обучения стартовой модели 10-ю эпохами получаем обновлённый файл весов best.pt с метриками изображенными на рисунке 42.

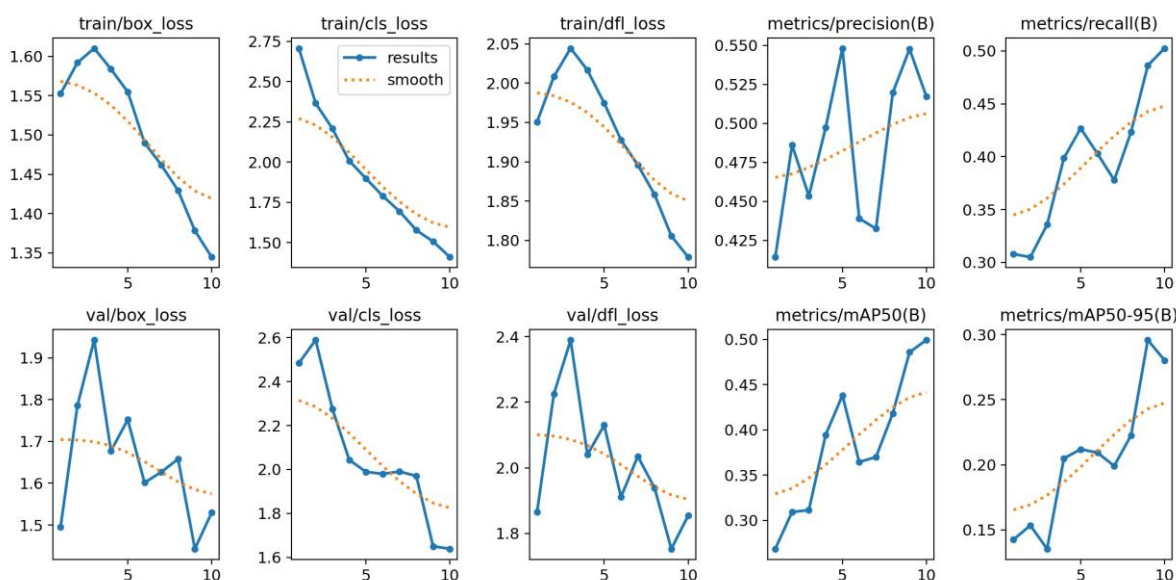


Рисунок 42 – Метрики обучения первых 10 эпох

После производим тестовое детектирование и проверяем на наличии в выходной видеозаписи артефактов и подозрительного поведения.

Далее приступаем к дообучению модели. В качестве датасета можно использовать как предыдущий датасет, так и другой подходящий с такой разметкой. Информация по сведению датасетов к одной аннотации рассказано в следующем пункте.

Для дообучения модели в функции обучения в качестве аргумента весов используем best.pt из предыдущего запуска.

В результате дообучения модели 15-ю эпохами получаем обновленный файл весов best.pt с метриками изображенными на рисунке 43.

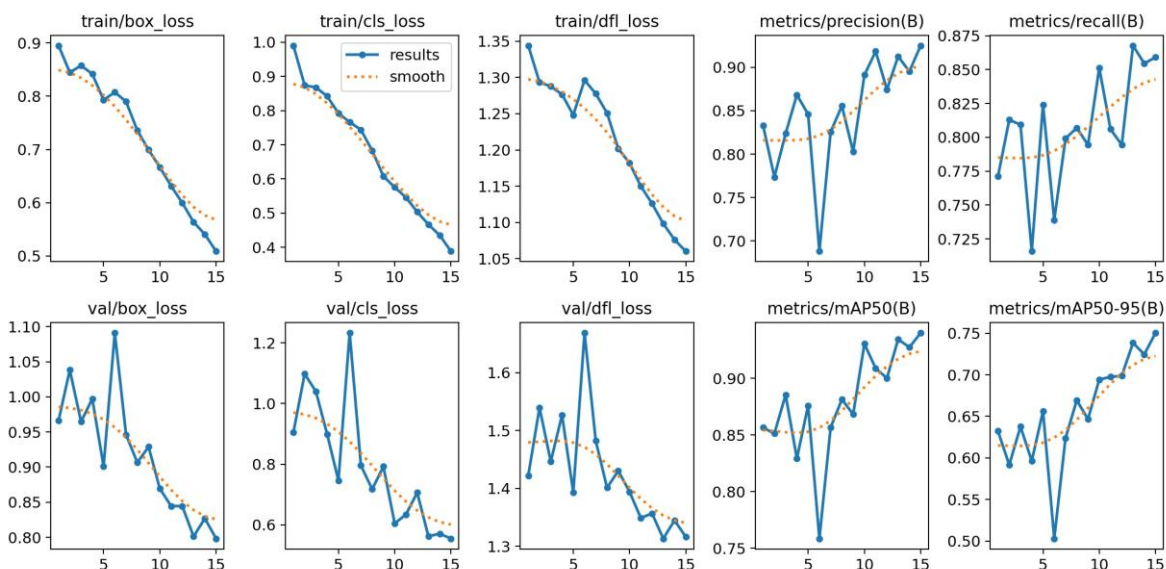


Рисунок 43 – Метрики дообучения 15 эпох

На рисунке 44 изображены детекции упаковок обновленной моделью.

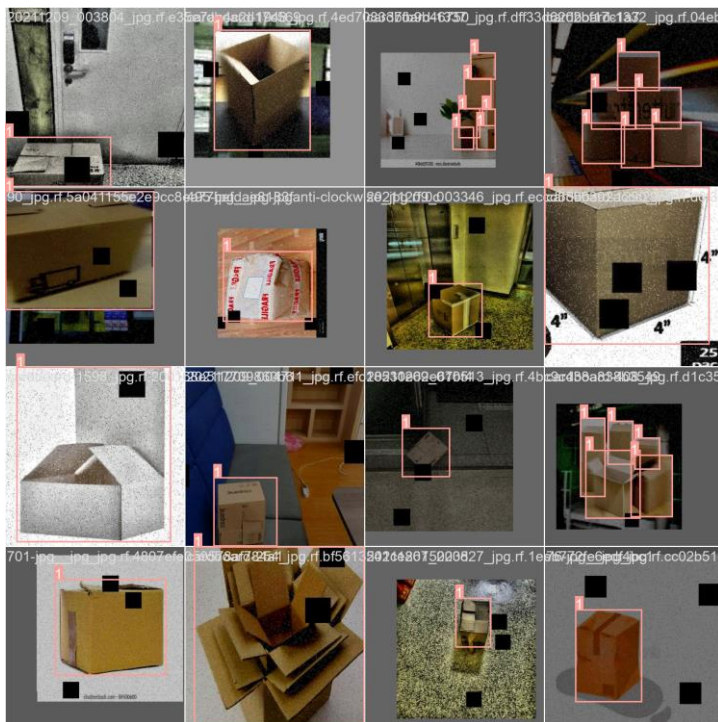


Рисунок 44 – Детекция дообученной модели

### 3.3.1.3 Приведение пользовательских датасетов к одной аннотации

Создание единого формата аннотаций для пользовательских датасетов является одним из ключевым шагов в процессе дообучения модели. Разнообразие

форматов аннотаций в различных источниках данных может привести к несоответствиям и затруднениям в процессе обучения. Стандартизация аннотаций упрощает процесс интеграции данных из разных источников и повышает качество обучения модели. Это может включать унификацию форматов ограничивающих рамок, меток классов и других параметров, что позволяет создать согласованный и высокоэффективный датасет для обучения и дообучения моделей компьютерного зрения.

Таким образом, эффективное решение задачи детекции на складах с помощью YOLOv8, адаптация и стандартизация данных обеспечивают значительные преимущества для автоматизации и оптимизации логистических процессов, повышая точность работы системы управления складом и снижая вероятность ошибок.

Для корректного последовательного обучения на нескольких датасетах требуется привести их к одной аннотации. Обратимся к `yaml` файлам в которых указаны пути к данным, количество и имена классов и прочая информация необходимая для начала обучения.

Наблюдаем что в файлах из листингов 1 и 2, разное количество классов и названия их различны. На это стоит обратить внимание так как это может привести к некорректному обучению что может пагубно повлиять на всю цепочку обучения.

## Листинг 1 – Содержание файла dataset1.yaml

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 2
names: ['label', 'package']

roboflow:
  workspace: larry-motalavigne
  project: package-and-label-detector
  version: 3
  license: CC BY 4.0
```

## Листинг 2 – Содержание файла dataset2.yaml

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 1
names: ['0']

roboflow:
  workspace: ieeeroboticswsu
  project: box-detector-weefm
  version: 1
  license: CC BY 4.0
```

Еще стоит обратить внимания что в первом датасете класс label стоит под номером 0, класс package под номером 1, а во втором класс label отсутствует, а класс package имеет название «0» и находится он под номером 0. Это значит что помимо аннотации требуется еще поменять набор лейблов изображения находящихся в папках имеющих «../train/labels», «../valid/labels», «../test/labels».

Для корректного изменения классов координат в файлах из папок labels используется скрипт на языке Python из листинга 3.

### Листинг 3 – Скрипт для изменения индексов координат в файлах images.txt

```
import os
# На вход подается путь к директории изменений
def modify_files_in_directory(directory):
    for filename in os.listdir(directory):
        file_path = os.path.join(directory, filename)

        # Проверяем, является ли файл обычным файлом
        if os.path.isfile(file_path):
            with open(file_path, 'r') as file:
                lines = file.readlines()

            with open(file_path, 'w') as file:
                for line in lines:
                    # Меняем первый символ с '0' на '1', если он существует
                    if line.startswith('0'):
                        line = '1' + line[1:]
                    file.write(line)
    print("Успешно изменено в " + directory)
```

После редактирования классов координат в файла labels, закончим изменения yaml файла второго датасета.

### Листинг 4 – Обновленное содержание файла dataset2.yaml

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 2
names: ['label', 'package']

roboflow:
    workspace: ieeeroboticswsu
    project: box-detector-weefm
    version: 1
    license: CC BY 4.0
```

Теперь можно производить последовательное дообучение с произвольным чередованием датасетов. Результаты детекции обученной модели изображено на рисунке 45.



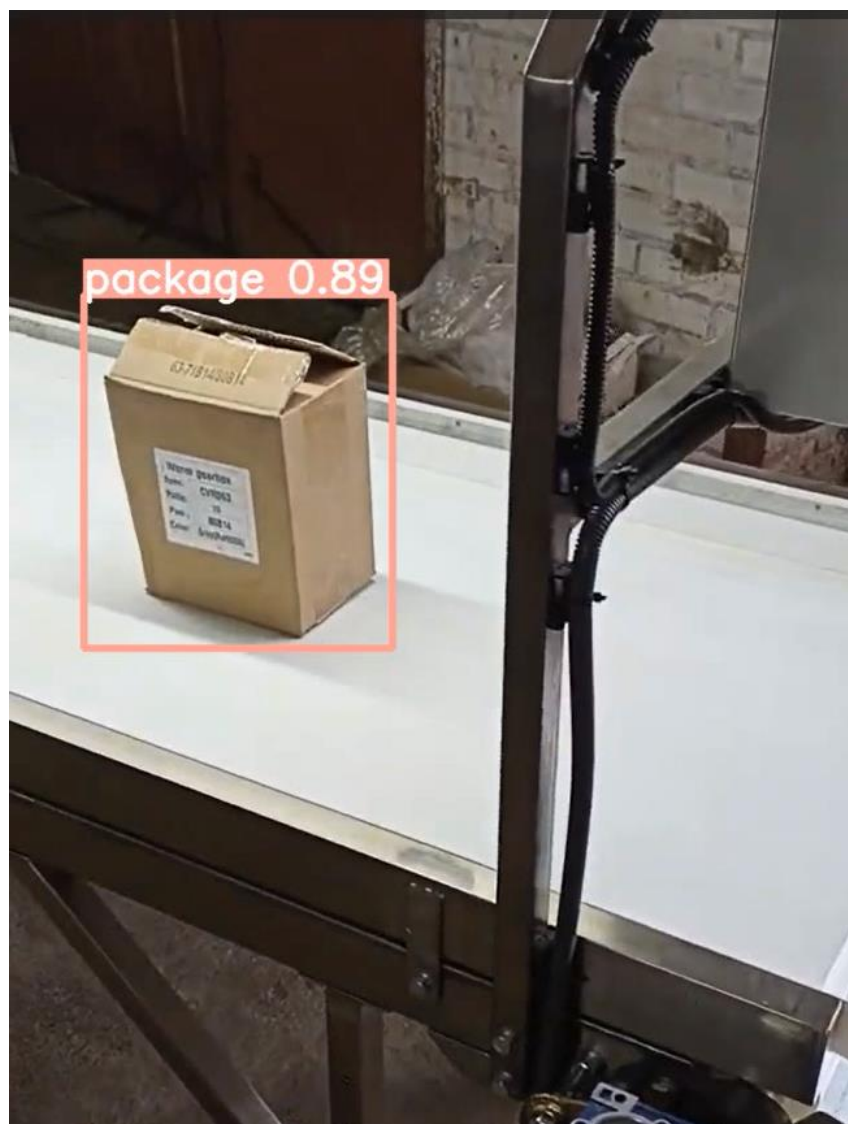


Рисунок 45– Результаты детекции

### 3.3.2 Решение задачи сегментации

Сегментация изображений представляет собой процесс разделения цифрового изображения на несколько сегментов (набор пикселей, также называемых суперпикселями) для упрощения или изменения представления изображения для улучшения его анализа. Сегментация целевых объектов на изображении может существенно улучшить последующие задачи, такие как распознавание объектов, трекинг и классификация, что, в свою очередь, способствует повышению эффективности автоматических систем управления на складах. Позволяет в дальнейшем определить габаритные характеристики груза.

В таблицах 3, 4 представлены метрики популярных моделей сегментации.

Таблица 3 – Показатели производительности YOLOv8 для задач сегментации

Модель	Размер (пикселей)	mAPbox 50-95	mAPmask 50-95	Скорость CPU ONNX(мс)	Скорость A100 TensorRT(мс)	params(M)	FLOPs(B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Таблица 4 – Показатели производительности YOLOv8 и SAM

Модель	Размер	Параметры	Скорость (процессор)
Meta's SAM-b	358 МБ	94.7 М	51096 ms/im
MobileSAM	40,7 МБ	10.1 М	46122 ms/im
FastSAM-s с магистралью YOLOv8	23,7 МБ	11.8 М	115 ms/im
Ultralytics YOLOv8n-seg	6,7 МБ	3,4 М	59 ms/im

Стоит обратить внимание что при сравнении наиболее упрощенных и быстродействующих моделей, показатели YOLOv8n-seg многократно обгоняют своих оппонентов, например, размер в 53,4 раза меньше, параметры в 27,9 раз меньше, а время обработки одного образа в 866 раз быстрее.

### 3.3.2.1 Применение модели SAM для сегментации объектов

Среди различных доступных подходов к сегментации объектов на изображениях одним из точных является применение модели Segment Anything Model (SAM), которая способна выделять почти любые объекты в различных условиях освещения и фона. Эта гибкость особенно важна в складских условиях, где товары могут иметь разнообразные формы, размеры и располагаться в сложно структурированных средах. Модель SAM использует методы глубокого обучения, сверточные нейронные сети, что позволяет ей обучаться на основе аннотированных изображений и выделять семантические границы интересующих объектов. Данная модель может быть интегрирована с системами компьютерного зрения на складе для выполнения задач сортировки, инвентаризации и управления запасами.

Сегментация является одним из ключевых этапов в обработке изображений и компьютерном зрении и позволяет:

- 1) Улучшить эффективность Обработки. Сегментация уменьшает объем обрабатываемых данных, выделяя важные объекты и их контуры, что делает последующий анализ более эффективным.
- 2) Распознавание объектов. Она помогает в точном определении положения и формы различных объектов на изображении, что важно для их распознавания и классификации.
- 3) Анализ изображений. Сегментация может использоваться для анализа медицинских изображений, аэрофотоснимков, исследований по биологии, метеорологии и многих других областей.
- 4) В робототехнике сегментация используется для интерпретации сенсорных данных для облегчения процессов принятия решений.
- 5) Изменение и улучшение изображений. Сегментация может использоваться для редактирования изображений, например, для изменения фона или для более детальной обработки определенных участков изображения.
- 6) Трекинг объектов. В видеопотоках сегментация позволяет отслеживать движение объектов.
- 7) В автономных системах, таких как беспилотные автомобили, сегментация помогает в интерпретации и понимании сцен в реальном времени для навигации и взаимодействия с окружающей средой.

Пример сегментированного изображения SAM представлен на рисунке 46.

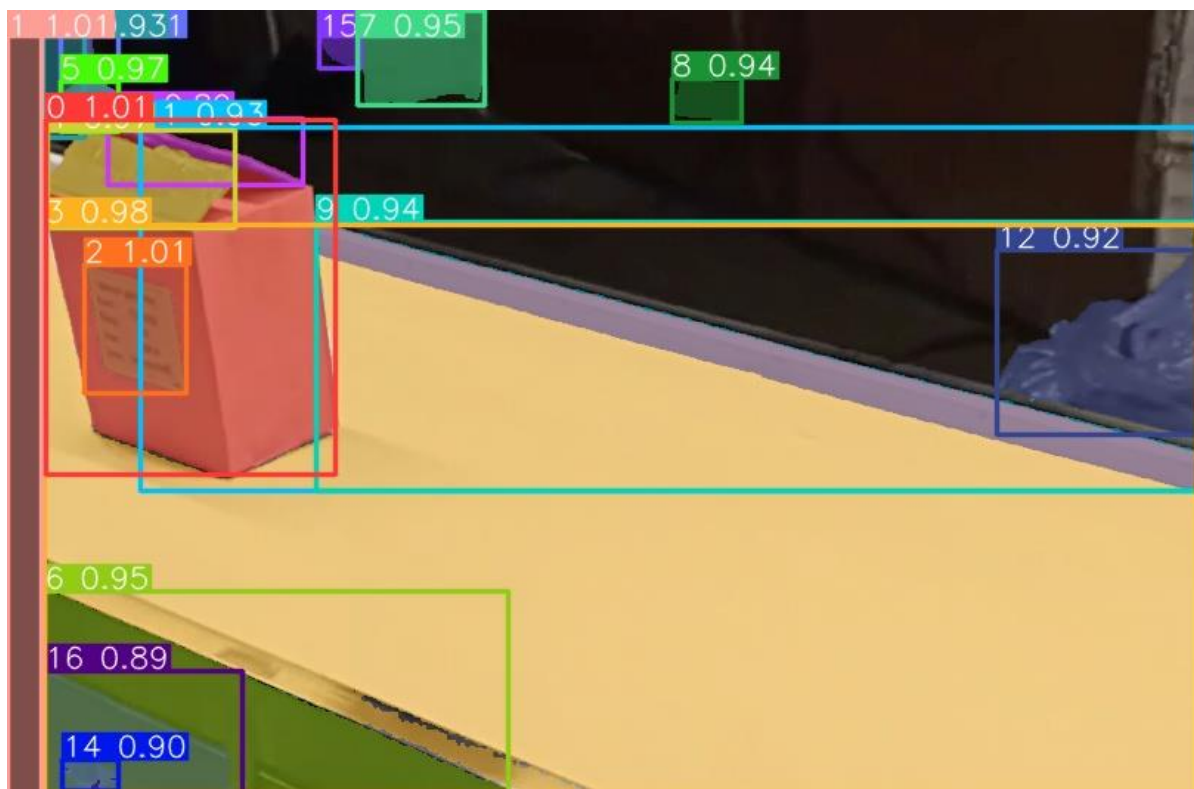


Рисунок 46 – Пример сегментации SAM

Также в процессе сегментации можно выделить и вырезать каждый слой, это позволяет в дальнейшем обрабатывать изображение по кускам, пример фрагмента изображен на рисунке 47, небольшой фрагмент более удобно обрабатывать, производить расчеты и разделять на более мелкие фрагменты, например для нахождения габаритных характеристик.



Рисунок 47 – Фрагмент сегментированного изображения

### 3.3.2.2 Создание набора данных сегментации, используя модель обнаружения

В процессе разработки систем компьютерного зрения часто возникает необходимость в создании наборов данных для специфических задач, таких как сегментация изображений. Однако, сбор и аннотация данных для сегментации может быть трудоемким процессом. Рассмотрим подход, который позволяет автоматизировать создание датасета для сегментации на основе существующего датасета, предназначенного для задачи детекции. Используя модель обнаружения объектов YOLOv8 и модель сегментации SAM (Segment Anything Model), предлагается метод автоматической аннотации, который значительно упрощает и ускоряет процесс подготовки данных.

Процесс начинается с использования модели YOLOv8, обученной на датасете для детекции. Эта модель способна эффективно идентифицировать и локализовать объекты на изображениях, предоставляя ограничивающие рамки вокруг каждого обнаруженного объекта. Однако, для задачи сегментации требуются более детальные аннотации, включающие пиксельные маски, которые точно определяют границы объектов. Здесь на помощь приходит модель SAM, которая способна генерировать такие маски на основе областей, выделенных YOLOv8.

Используя функцию `auto_annotate` из библиотеки `Ultralytics`, процесс преобразования датасета для детекции в датасет для сегментации автоматизируется. Эта функция принимает путь к изображениям (`path/to/images`), модель YOLOv8 (`yolov8x.pt`), и модель SAM (`sam_b.pt`). Алгоритм проходит через каждое изображение в исходном датасете, использует YOLOv8 для обнаружения объектов, а затем применяет SAM для создания точных пиксельных масок для каждого обнаруженного объекта. Результатом является датасет, содержащий изображения с соответствующими масками сегментации, который может быть использован для обучения и оценки моделей сегментации.

На рисунке 48 изображен пример сегментации для модели YOLOv8l-seg.

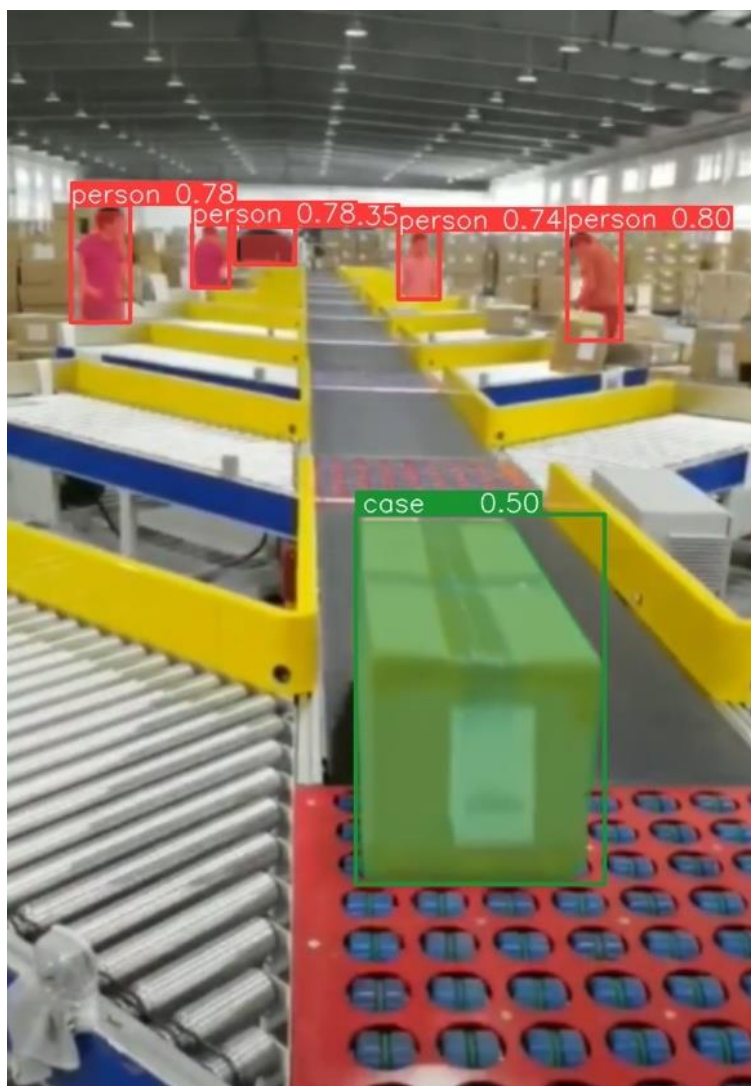


Рисунок 48 – Пример сегментации модели YOLOv8-seg

Такой подход значительно сокращает время и усилия, необходимые для подготовки датасетов сегментации, и предлагает эффективный способ повторного использования и расширения существующих датасетов для детекции. Кроме того, он позволяет создавать высококачественные наборы данных сегментации, которые могут быть адаптированы к различным условиям и требованиям конкретных приложений компьютерного зрения.

### 3.3.3 Создание синтетической тестовой среды в редакторе трехмерной графики Blender

Для разработки и валидации модели компьютерного зрения для детекции и сегментации товаров на сортировочной ленте была создана синтетическая тестовая среда в редакторе трехмерной графики Blender. Этот процесс включал несколько важных этапов.

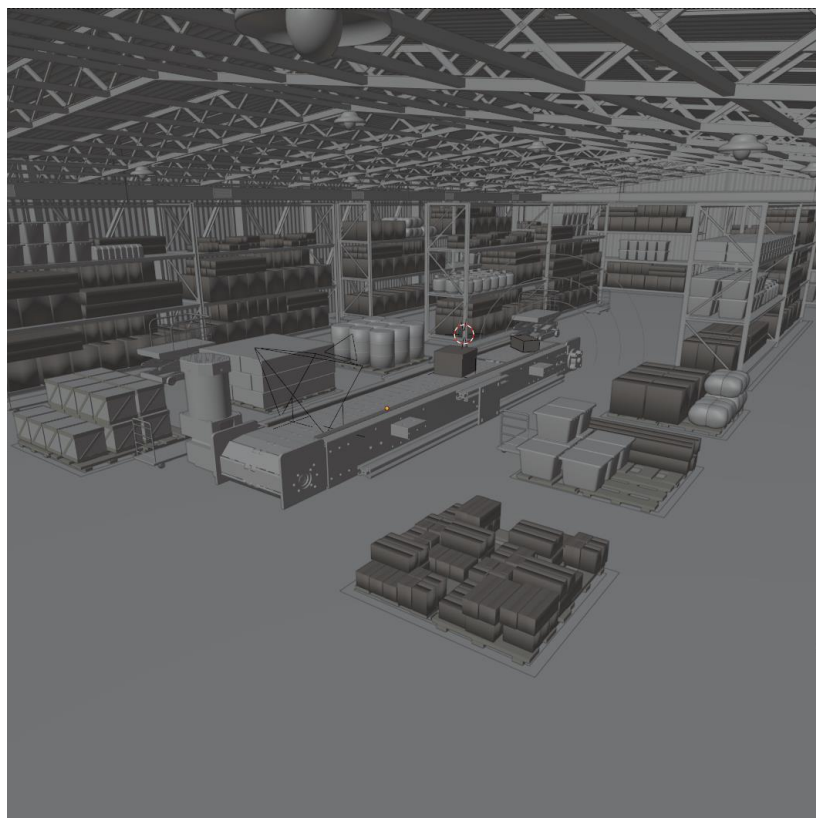


Рисунок 49 – Модель тестового склада

В начале были созданы объекты окружения, использовались готовые модели из открытых источников, собрано несколько моделей складских помещений, конвейеров и объектов имитирующие товары, которые будут перемещаться по сортировочной ленте. Для этих объектов были определены параметры коллизий, задана их масса, а также параметры твердого тела в динамике.



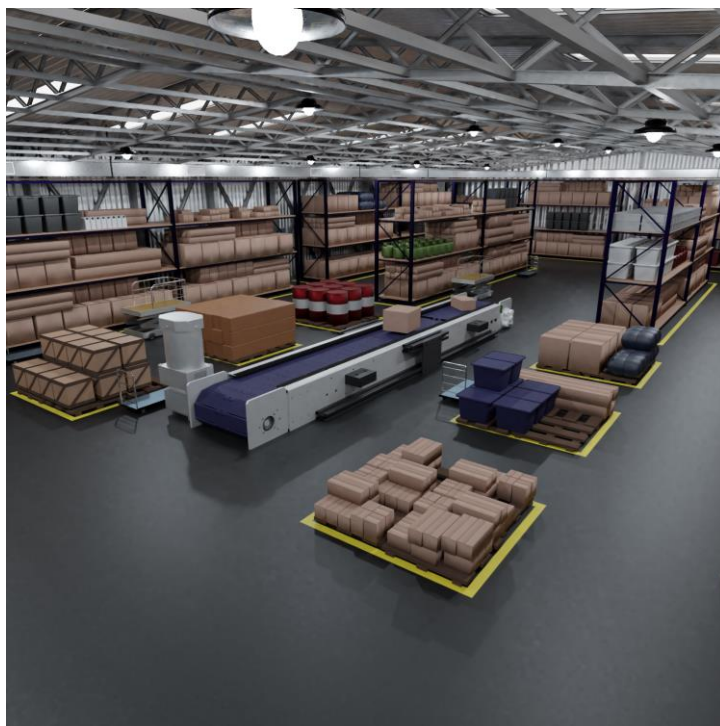


Рисунок 50 – Рендер рабочего тестового пространства

Для эмуляции движений объектов по ленте был установлен "источник силы" в заданном направлении движения ленты. Также были подобраны параметры камеры для захвата изображений с оптимальным ракурсом и разрешением.

Для получения разнообразных сценариев движения была собран видеоряд, включающий несколько сотен кадров.

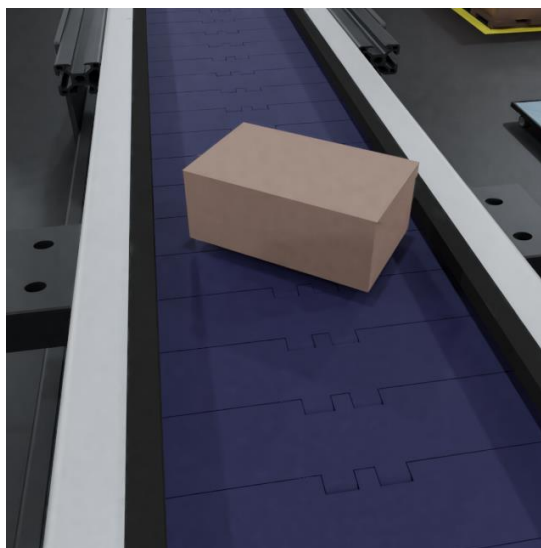


Рисунок 51 – Кадр из видеоряда записанного на рабочую камеру

Для эффективной работы модели необходимо было провести калибровку рабочей области. Этот процесс включал в себя измерение выбранной



прямоугольной области ленты и перенос меток в координаты пикселей на изображении с камеры. Для автоматизации этого процесса были установлены параллельные метки на краях бортов конвейера, и затем проведено маркирование на изображении. Это позволило интерпретировать данные с изображений в реальные размеры с помощью перспективных преобразований.

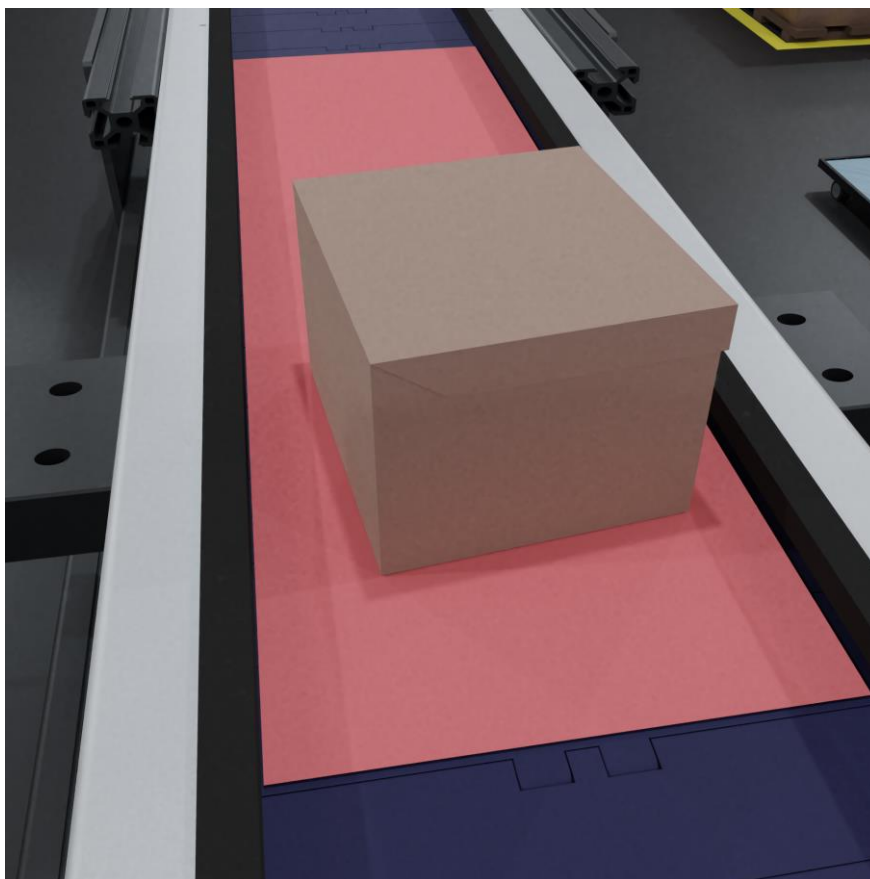


Рисунок 52 – Выделение рабочей откалиброванной области

Созданная синтетическая тестовая среда предоставляет возможность генерировать данные для обучения модели компьютерного зрения. Она также позволяет проводить валидацию модели в реалистичных условиях работы системы, что способствует улучшению качества обучения и эффективности работы модели компьютерного зрения в реальном мире.

#### 3.3.4 Алгоритм определения облака точек объекта

Алгоритм определения облака точек объекта представляет собой процесс, включающий детекцию объекта на сортировочной ленте и последующую сегментацию его на детали. Этот процесс основан на идее проведения детекции

один раз и сегментации два раза, что позволяет эффективно обработать изображения и выделить различные части объекта.

В начале производится детекция объекта на ленте. Из данных детекции определяются координаты ограничивающего прямоугольника (bbox) объекта. Этот этап помогает отфильтровать ненужные объекты и упростить дальнейшую сегментацию. На левом изображении рисунка 53 представлен bbox детекции.

После этого происходит сегментация изображения на основе данных детекции. Определяется единственный объект, и по данным сегментации получается набор точек, представляющих криволинейный контур объекта. Эти точки нужно разделить на грани объекта для дальнейшей обработки. На правом изображении рисунка 53 представлен пример сегментации цельного объекта.

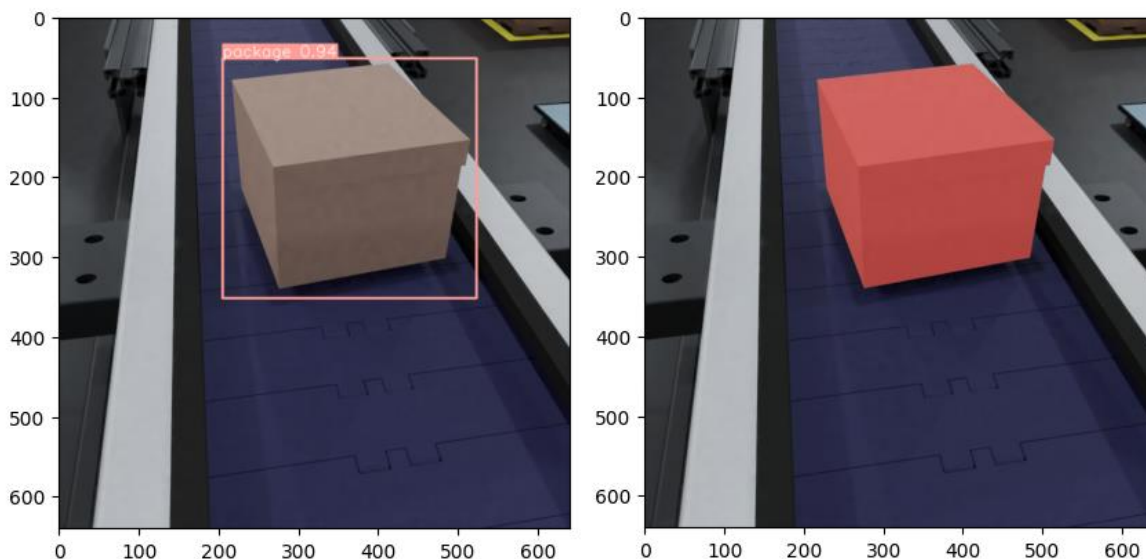


Рисунок 53 – Пример детекции и сегментации объекта

Для сегментации граней объекта требуется предварительная подготовка, включающая обрезку изображения и помещение объекта на контрастный фон. Опционально используется белый или черный, чтобы обеспечить максимальный контраст. Создается маска объекта из точек, полученных на предыдущем этапе

сегментации, для эффективного удаления фона и минимизации ложной сегментации лишних объектов. На рисунке 54 представлено подготовленное изображение для детальной сегментации.

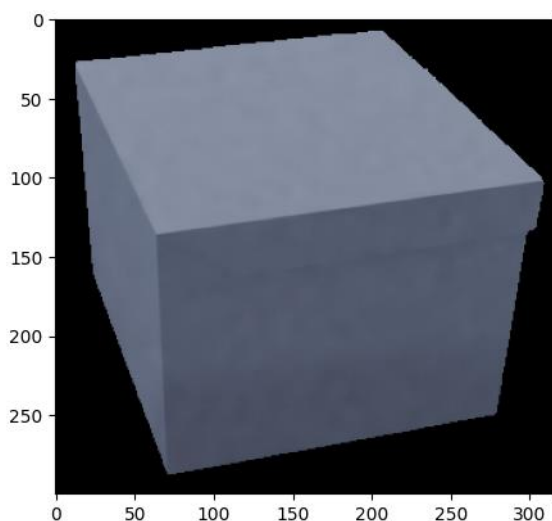


Рисунок 54 – Подготовленное изображение для детальной сегментации

Далее требуется провести детальную сегментацию, тут есть несколько эффективных подходов, такие как детальная сегментация моделью Segmentation Anything и наложение фильтров, и поиск контуров OpenCV.

#### 3.3.4.1 Детальная сегментация Segmentation Anything

Затем проводим повторную сегментацию. В результате исходный объект по маске сегментируется на несколько объектов, которые представляют собой грани упаковки товара. Каждый набор точек относится к своей грани объекта, что позволяет получить полностью классифицированное облако точек объекта. На рисунке 55 представлена детальная сегментация целевого объекта.

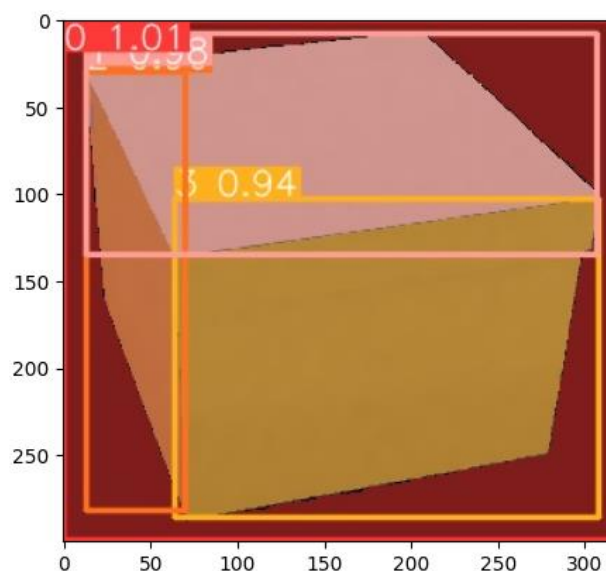


Рисунок 55 – Детальная сегментация целевого объекта

В результате получаем несколько наборов точек, каждый из набор относится к своей грани упаковки товара. Алгоритм обеспечивает эффективное выделение всех деталей объекта на сортировочной ленте и позволяет использовать полученные данные для дальнейшего анализа и обработки.

На рисунке 56 изображены облака точек, точки граней выделены отдельными цветами, также на изображении присутствуют линии границы откалиброванной рабочей области.

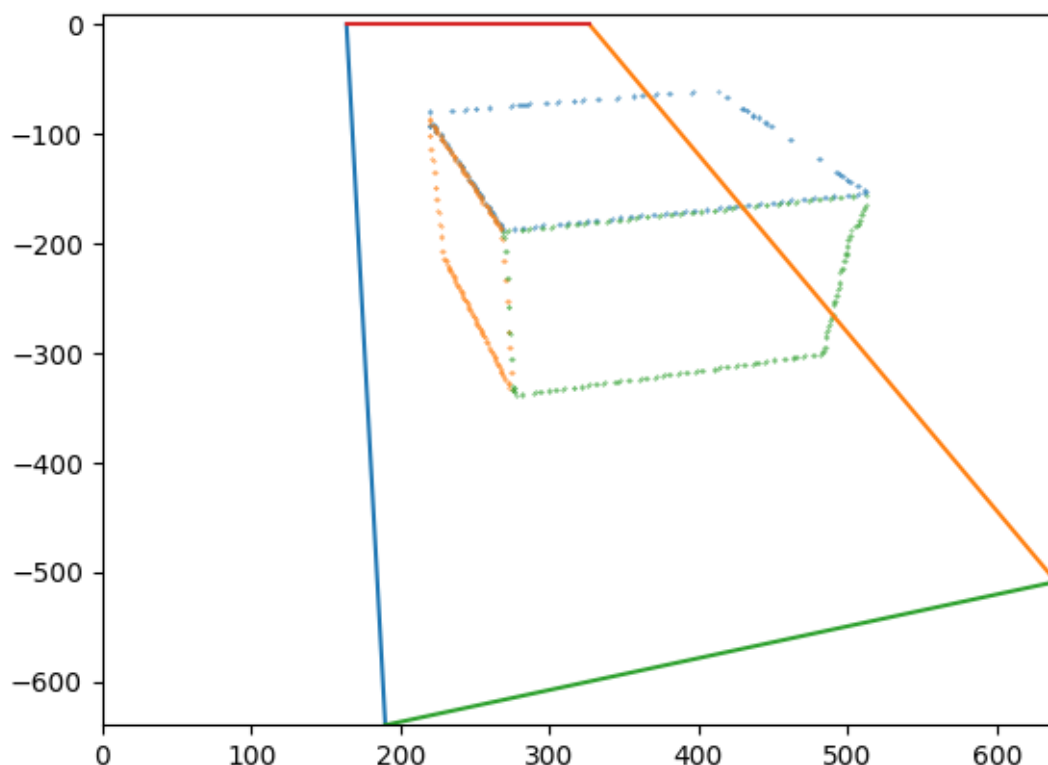


Рисунок 56 – Классифицированные облака точек в рабочей зоне

#### 3.3.4.2 Детальная сегментация OpenCV

В рамках данного метода обработки изображений первоначально применяется оператор Собеля к предварительно обработанному изображению. Этот оператор основан на дифференциальном вычислении градиента яркости в изображении и направлен на выявление изменений яркости в горизонтальном и вертикальном направлениях.

Далее рассчитывается величина градиента, используя значения, полученные после применения оператора Собеля как по горизонтали, так и по вертикали.

Затем, полученная величина градиента используется для проведения бинаризации изображения с использованием порогового значения. Пороговое значение задается изначально при калибровке камеры, также возможно адаптивное изменение порогового значения в зависимости от критериев, заданных итоговому изображению.

Этот шаг позволяет разделить изображение на области, соответствующие различным степеням изменения яркости. В результате получается бинарное

изображение, где пиксели с интенсивностью выше порогового значения классифицируются как объекты, а остальные пиксели воспринимаются как фон.

На рисунке 57 изображено подготовленное изображение после применения фильтра Собеля.

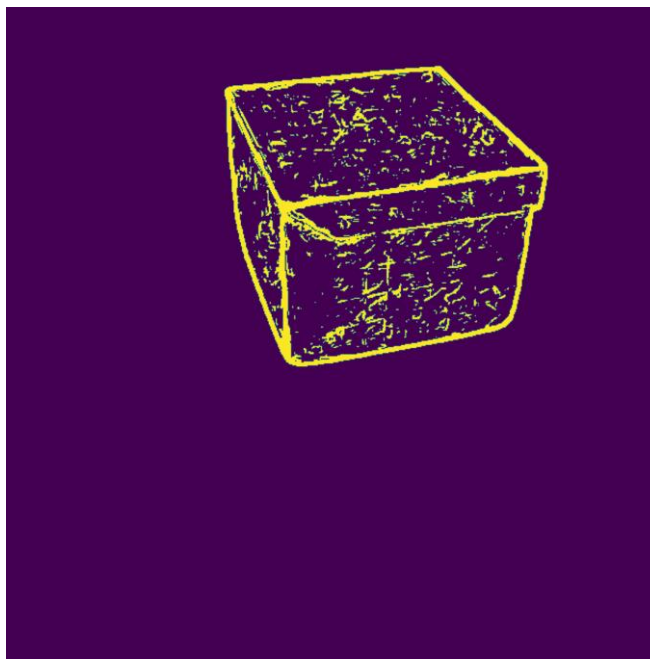


Рисунок 57 – Фильтр Собеля

Итоговой операцией сегментации является поиск контуров на бинарном изображении. Контуров представляют собой области, где интенсивность пикселей изменяется значительно, что позволяет выявить формы и структуры объектов на изображении.

На выходе имеется набор контуров, которые требуется отфильтровать. Лишними контурами являются элементы, которые не являются замкнутыми или слишком крохотными. Определение нужных контуров осуществляется сортировкой по количеству точек и площади замкнутых контуров.

Этот этап позволяет получить границы объектов и их структурные особенности для дальнейшего анализа и обработки.

На рисунке 58 представлен отфильтрованный набор контуров, который далее преобразуется в набор точек.

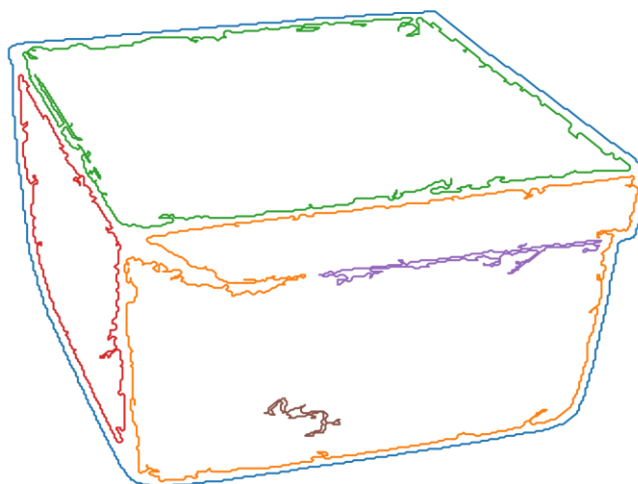


Рисунок 58 – Набор контуров сегментированного изображения

### 3.3.5 Преобразование облака точек в детализированные элементы объекта

Преобразование облака точек в детализированные элементы объекта начинается с того, что на входе имеется несколько наборов точек для каждой грани объекта. Наша задача состоит в том, чтобы аппроксимировать эти облака точек и преобразовать их в грани. Стремимся к тому, чтобы каждая грань объекта была представлена простыми фигурами с минимальным количеством точек. Для этого используется метод аппроксимации, который позволяет нам уменьшить количество точек до определенного числа, например, до четырех точек на грань.

Наиболее подходящий метод аппроксимации включает комбинацию метода Рамера-Дугласа-Пекера и локальной интерполяции. Метод Рамера-Дугласа-Пекера используется для уменьшения количества точек в кривой, сохраняя ее форму. Он работает путем рекурсивного удаления «лишних» точек, которые могут быть аппроксимированы без значимой потери качества. Локальная интерполяция используется для добавления новых точек там, где это необходимо, чтобы

гарантировать, что аппроксимация соответствует форме исходной грани с высокой точностью.

Алгоритм аппроксимации следующий:

- 1) На входе у нас есть список из  $n$ -го количества двумерных массивов с координатами точек для каждой грани объекта.
- 2) Находим минимальный по площади многоугольник, в который вписываются все точки из списка.
- 3) Применяем метод Рамера-Дугласа-Пекера к каждому описанному полигону, чтобы уменьшить количество точек до определенного порога.
- 4) После этого итеративно применяем локальную интерполяцию к каждому минимальному ребру, чтобы ликвидировать ненужные грани.
- 5) Задать новые точки там, где это необходимо и сохранить форму общей грани.

В результате получаем детализированные элементы объекта, представленные в виде граней с углами, наиболее точно отражающими форму исходного объекта.

Этот метод аппроксимации обеспечивает эффективное преобразование облака точек в детализированные элементы объекта, что позволяет получить более точное представление структуры объекта и облегчает дальнейший анализ и обработку данных.

На рисунке 59 представлено несколько способов аппроксимации облака точек.



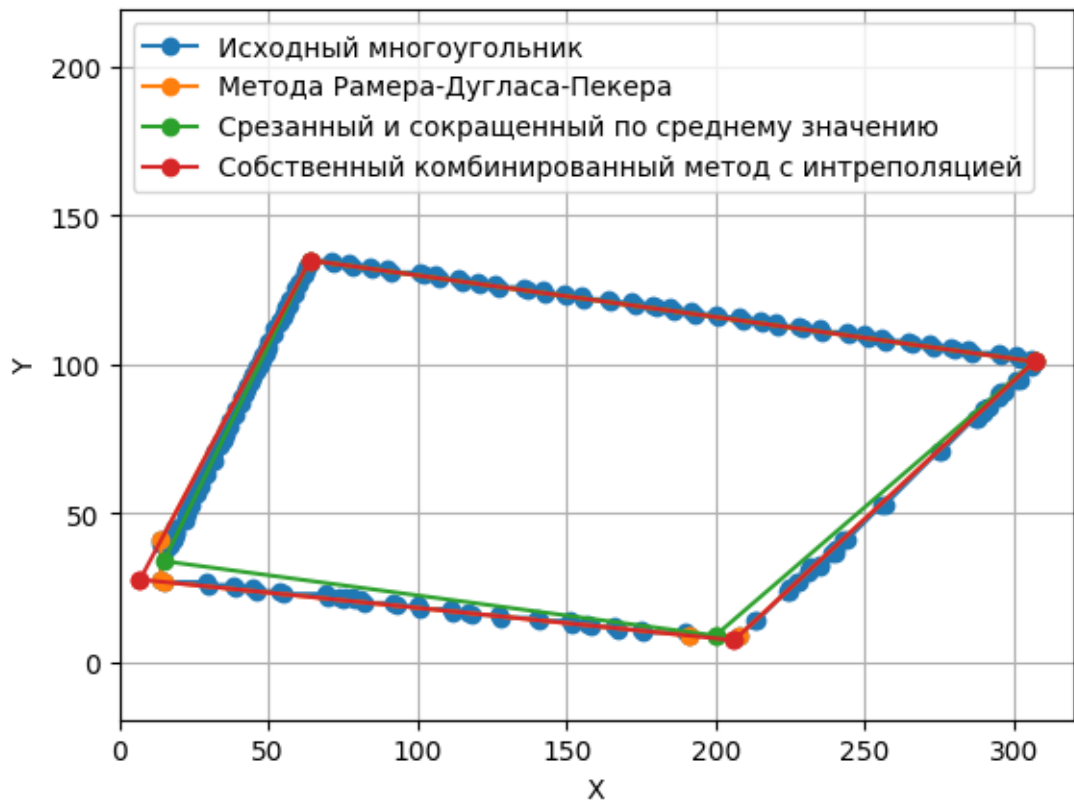


Рисунок 59 – Сравнение методов аппроксимации

В результате аппроксимации получаем несколько массивов точек, которые нужно преобразовать в удобный для передачи вид.

Преобразуем данные наборы в формат json.

Листинг 4 – Фрагмент представления данных в json

```
{'sides': [{'edges': [{'line': {'xy': [[411.40599614753864, -
59.52388654678937], [212.08042401421156, -
79.90945642406146]]}, {'centoroid': {'xy': [311.7432100808751, -
69.71667148542542]}]}],
{'line': {'xy': [[212.08042401421156, -
79.90945642406146], [269.0914306640625, -
187.09014892578125]]}, {'centoroid': {'xy': [240.58592733913704, -
133.49980267492134]}]}],
{'line': {'xy': [[269.0914306640625, -
187.09014892578125], [512.0914306640625, -
153.09014892578125]]}, {'centoroid': {'xy': [390.5914306640625, -
170.09014892578125]}]}],
{'line': {'xy': [[512.0914306640625, -
153.09014892578125], [411.40599614753864, -
59.52388654678937]]}, {'centoroid': {'xy': [461.7487134058006, -
106.3070177362853]}]}],
{'edges': [{'line': {'xy': [[217.33673057453697, -
```

На рисунке 60 изображен результат аппроксимации набора точек на рабочем изображении.

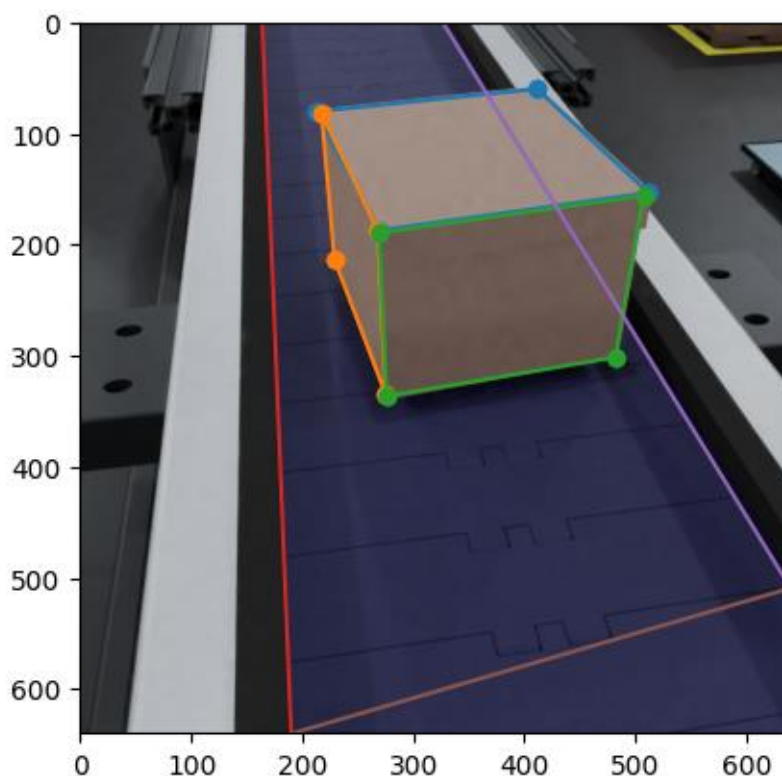


Рисунок 60 – Аппроксимированные наборы точек

### 3.3.6 Расчет габаритных характеристик товара на ленте

Расчет габаритных характеристик товара на сортировочной ленте начинается с определения известных координат границ рабочей области и реальных размеров откалиброванной рабочей области. Идея состоит в том, чтобы преобразовать исходное изображение, полученное с камеры, в подобие ортогонального изображения, в котором размеры нижней грани товара будут совпадать с физическими параметрами объекта в реальной жизни. Для этого необходимо сопоставить координаты на изображении и реальные размеры как параметры преобразования в матрицу перспективы. Затем с помощью полученной матрицы перспектив преобразуем изображение с камеры.

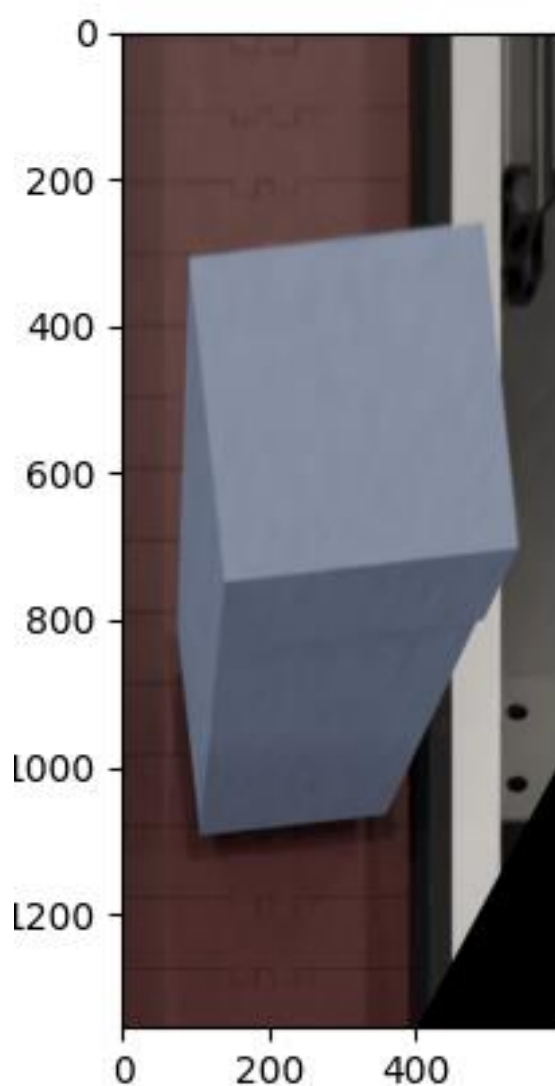


Рисунок 61 – «Ортогональное» изображение рабочей области и объектов на ней

Поскольку данные ребер сгруппированы по граням, есть возможность ориентировать их в пространстве. Сначала находим самое верхнее и нижнее ребро в наборе по ординате, чтобы определить, какая грань является наиболее "лицевой" и наиболее "верхней". Затем ориентируем ребра лицевой и верхней грани в пространстве.

Для нахождения ширины, то есть нижней лицевой грани, измеряем длину между точками на «ортогональном» изображении.

Для нахождения длины требуется узнать коэффициент увеличения, которым является отношение длин верхнего и нижнего ребра лицевой грани. Затем аналогично ширине узнаем длину верхней грани, умножив ее на коэффициент увеличения.

Для расчета высоты необходимо определить размер левого/правого ребра лицевой стороны, умножив его на коэффициент увеличения, полученный при расчете длины. Для повышения точности можно дифференцировать это ребро относительно коэффициента увеличения, разложив его на более мелкие отрезки и учитывая нелинейную зависимость коэффициента увеличения.

Таким образом, получаем основные параметры объекта на ленте. После проведения всех вычислений и преобразований отображаем габаритные характеристики на исходном изображении, что поможет в дальнейшем анализе и обработке данных.

Пример отображения габаритных характеристик на исходном изображении.

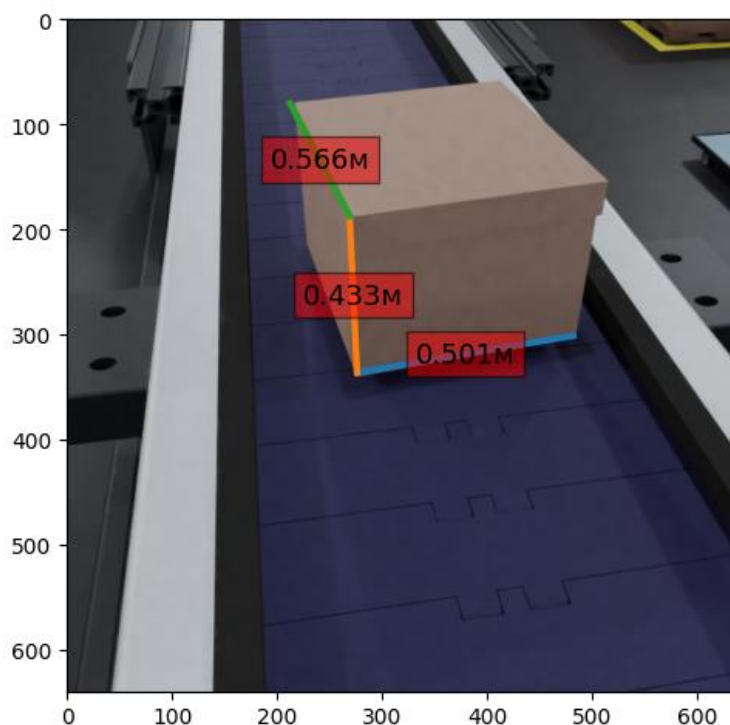


Рисунок 62 – Отображение габаритных характеристик на исходном изображении

### 3.4 Вывод к главе

В рамках данной главы были изучены и адаптированы методы повышения точности детектирования объектов на изображениях в сфере складской логистики. Рассмотрены улучшения предварительной обработки изображений, оптимизации гиперпараметров и функций потерь, а также применение синтетических

и дополненных данных позволило углубить понимание влияния этих аспектов на эффективность моделей глубокого обучения.

Особое внимание было уделено применению модели YOLOv8 для задач обнаружения объектов, что демонстрирует значительные перспективы в повышении скорости и точности детекции в реальных условиях работы склада. Разработка методологии дообучения пользовательской модели и стандартизация пользовательских датасетов под одну аннотацию позволили не только адаптировать модели под конкретные требования, но и значительно повысить их обрабатывающую способность.

В области сегментации было установлено, что использование модели SAM в тандеме с YOLOv8 является эффективным решением для точного выделения объектов на изображениях, что открывает новые возможности для улучшения процессов управления на складе. Также была рассмотрена возможность создания набора данных сегментации с использованием модели обнаружения, что обеспечивает более высокую точность и универсальность в разработке систем компьютерного зрения.

В данной главе были рассмотрены ключевые этапы разработки системы компьютерного зрения для детекции и сегментации товаров на сортировочной ленте. В частности, описаны процессы создания синтетической тестовой среды, определения облака точек объекта, их преобразования в детализированные элементы, а также расчета габаритных характеристик товара на ленте. Каждый этап представляет собой важный шаг в разработке системы и включает в себя множество технических деталей и методов. В результате выполнения этих этапов достигается создание системы, способной эффективно обрабатывать изображения, выделять объекты и определять их основные характеристики. Это позволяет улучшить качество обучения модели компьютерного зрения и повысить эффективность ее работы в реальных условиях.

В заключение, достигнутые результаты подчеркивают значимость интеграции современных технологий компьютерного зрения в логистические процессы и открывают пути для дальнейших исследований в этом направлении.

Предложенные подходы и разработанные методики способствуют повышению эффективности, надежности и точности автоматизированных систем складской логистики, что имеет важное значение для динамично развивающейся сферы современной индустрии.

## ЗАКЛЮЧЕНИЕ

Исследование подтвердило, что интеграция компьютерного зрения в складскую логистику предоставляет значительные преимущества для оптимизации процессов управления и повышения операционной эффективности. Анализ современных проблем в области складской логистики выявил необходимость в поиске новых технологических решений для справления с увеличивающимся объемом грузов и ускоренными темпами работы. Компьютерное зрение представляет собой мощный инструмент для автоматизации складских операций, позволяя автоматически распознавать и сегментировать товары на основе изображений, контролировать качество упаковки и определять габаритные характеристики.

Разработка системы компьютерного зрения для складской логистики включала в себя полный цикл разработки, начиная от анализа требований и концептуального проектирования, и заканчивая созданием тестовой среды и проведением экспериментальных исследований. Успешное детектирование и сегментация товаров на сортировочной ленте подтверждает эффективность и применимость предложенного подхода. Полученные данные о товарах могут быть использованы для дальнейшего анализа и оптимизации складских процессов, что способствует повышению эффективности использования складских ресурсов и улучшению обслуживания клиентов. В целом, интеграция компьютерного зрения в складскую логистику представляет собой перспективное направление для современных складов, позволяя повысить их конкурентоспособность и эффективность в условиях быстро меняющегося рынка.