

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**  
**ITMO University**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**GRADUATION THESIS**

**Исследование и сравнение современных алгоритмов компьютерного зрения для отслеживания перемещения объектов на видеоизображениях в городской среде**

**Обучающийся / Student** Лалаянц Кирилл Артемович

**Факультет/институт/клластер/ Faculty/Institute/Cluster** факультет систем управления и робототехники

**Группа/Group** R34352

**Направление подготовки/ Subject area** 15.03.06 Мехатроника и робототехника

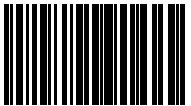
**Образовательная программа / Educational program** Робототехника и искусственный интеллект 2021

**Язык реализации ОП / Language of the educational program** Русский

**Квалификация/ Degree level** Бакалавр

**Руководитель ВКР/ Thesis supervisor** Шаветов Сергей Васильевич, доцент, кандидат технических наук, Университет ИТМО, факультет систем управления и робототехники, доцент (квалификационная категория "ординарный доцент")

Обучающийся/Student

Документ подписан	
Лалаянц Кирилл Артемович	
11.04.2025	

(эл. подпись/ signature)

Лалаянц  
Кирилл  
Артемович

(Фамилия И.О./ name  
and surname)

Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Шаветов Сергей Васильевич	
11.04.2025	

(эл. подпись/ signature)

Шаветов Сергей  
Васильевич

(Фамилия И.О./ name  
and surname)

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /  
OBJECTIVES FOR A GRADUATION THESIS**

**Обучающийся / Student** Лалаянц Кирилл Артемович

**Факультет/институт/клластер/ Faculty/Institute/Cluster** факультет систем управления и робототехники

**Группа/Group** R34352

**Направление подготовки/ Subject area** 15.03.06 Мехатроника и робототехника

**Образовательная программа / Educational program** Робототехника и искусственный интеллект 2021

**Язык реализации ОП / Language of the educational program** Русский

**Квалификация/ Degree level** Бакалавр

**Тема ВКР/ Thesis topic** Исследование и сравнение современных алгоритмов компьютерного зрения для отслеживания перемещения объектов на видеоизображениях в городской среде

**Руководитель ВКР/ Thesis supervisor** Шаветов Сергей Васильевич, доцент, кандидат технических наук, Университет ИТМО, факультет систем управления и робототехники, доцент (квалификационная категория "ординарный доцент")

**Характеристика темы ВКР / Description of thesis subject (topic)**

**Тема в области фундаментальных исследований / Subject of fundamental research:** нет / not

**Тема в области прикладных исследований / Subject of applied research:** да / yes

**Основные вопросы, подлежащие разработке / Key issues to be analyzed**

Техническое задание:

Исследование и сравнение современных алгоритмов компьютерного зрения для отслеживания перемещения объектов в видеопотоке в городской среде.

Цель:

Выполнить сравнительный анализ существующих подходов к отслеживанию объектов в видеопотоке с целью выявления наиболее подходящих для низкопроизводительных устройств алгоритмов на основе показателей производительности и метрик.

Задачи:

- 1) Аналитический обзор существующих подходов.
- 2) Подготовка стенда для апробации.
- 3) Подготовка эксперимента для сравнения различных алгоритмов на основании метрик НОТА, МОТА, IDF1 и показателей производительности.
- 4) Анализ полученных результатов.

Исходные данные к работе:

- 1) В качестве низкопроизводительного устройства использовать микрокомпьютер Raspberry Pi 5 с внешним вычислительным модулем TPU.
- 2) Рассмотреть не менее 3 различных алгоритмов.
- 3) Для сравнения использовать набор данных MOT17.

**Форма представления материалов ВКР / Format(s) of thesis materials:**

Пояснительная записка

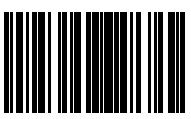
Презентация

**Дата выдачи задания / Assignment issued on:** 05.02.2025

**Срок представления готовой ВКР / Deadline for final edition of the thesis** 30.05.2025

**СОГЛАСОВАНО / AGREED:**

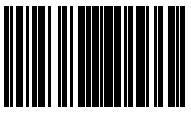
Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Шаветов Сергей Васильевич	
21.02.2025	

(эл. подпись)

Шаветов Сергей  
Васильевич

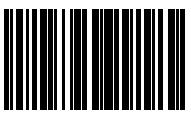
Задание принял к  
исполнению/ Objectives  
assumed BY

Документ подписан	
Лалаянц Кирилл Артемович	
21.02.2025	

(эл. подпись)

Лалаянц  
Кирилл  
Артемович

Руководитель ОП/ Head  
of educational program

Документ подписан	
Бобцов Алексей Алексеевич	
06.05.2025	

(эл. подпись)

Бобцов Алексей  
Алексеевич

**Министерство науки и высшего образования Российской Федерации**  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**  
**ITMO University**

**АННОТАЦИЯ**  
**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**  
**SUMMARY OF A GRADUATION THESIS**

**Обучающийся / Student** Лалаянц Кирилл Артемович

**Факультет/институт/кластер/ Faculty/Institute/Cluster** факультет систем управления и робототехники

**Группа/Group** R34352

**Направление подготовки/ Subject area** 15.03.06 Мехатроника и робототехника

**Образовательная программа / Educational program** Робототехника и искусственный интеллект 2021

**Язык реализации ОП / Language of the educational program** Русский

**Квалификация/ Degree level** Бакалавр

**Тема ВКР/ Thesis topic** Исследование и сравнение современных алгоритмов компьютерного зрения для отслеживания перемещения объектов на видеоизображениях в городской среде

**Руководитель ВКР/ Thesis supervisor** Шаветов Сергей Васильевич, доцент, кандидат технических наук, Университет ИТМО, факультет систем управления и робототехники, доцент (квалификационная категория "ординарный доцент")

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**  
**DESCRIPTION OF THE GRADUATION THESIS**

**Цель исследования / Research goal**

Сравнительный анализ существующих подходов к отслеживанию объектов в видеопотоке с целью выявления наиболее подходящих для низкопроизводительных устройств алгоритмов на основе показателей производительности и метрик

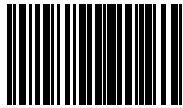
**Задачи, решаемые в ВКР / Research tasks**

1) Аналитический обзор существующих подходов. 2) Подготовка стенда для аprobации. 3) Подготовка эксперимента для сравнения различных алгоритмов на основании метрик НОТА, MOTA, IDF1 и показателей производительности. 4) Анализ полученных результатов.

**Краткая характеристика полученных результатов / Short summary of results/findings**

Проведен аналитический обзор шести алгоритмов отслеживания объектов на видеоизображениях. Подготовлен стенд для аprobации с использованием микрокомпьютера Raspberry Pi 5 с внешним вычислительным модулем TPU Google Coral. Проведено сравнение показателей качества алгоритмов на наборе данных MOT17 с учетом скорости работы. На основании проведенных экспериментов подготовлены методические рекомендации по выбору алгоритма.

подписан	
Лалаянц Кирилл Артемович	
11.04.2025	



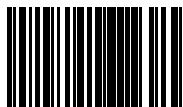
(эл. подпись/ signature)

Лалаянц  
Кирилл  
Артемович

(Фамилия И.О./ name  
and surname)

Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Шаветов Сергей Васильевич	
11.04.2025	



(эл. подпись/ signature)

Шаветов Сергей  
Васильевич

(Фамилия И.О./ name  
and surname)

# **СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ . . . . .</b>	8
<b>1 Задача отслеживания объектов на видеоизображениях . . . . .</b>	10
1.1. Постановка задачи . . . . .	10
1.2. Основные проблемы при решении задачи . . . . .	11
1.3. Актуальность . . . . .	12
1.4. Выводы по главе . . . . .	14
<b>2 Обзор современных подходов к отслеживанию объектов на видеоизображениях . . . . .</b>	15
2.1. Набор данных MOT17 . . . . .	15
2.2. Метрики для оценивания . . . . .	17
2.2.1. Метрика MOTA . . . . .	17
2.2.2. Метрика IDF1 . . . . .	18
2.2.3. Метрика HOTA . . . . .	19
2.3. Обзор существующих алгоритмов . . . . .	20
2.3.1. Алгоритм ByteTrack . . . . .	20
2.3.2. Алгоритм OC-SORT . . . . .	22
2.3.3. Алгоритм BoT-SORT . . . . .	23
2.3.4. Алгоритм StrongSORT . . . . .	24
2.3.5. Алгоритм ImprAssOC . . . . .	25
2.3.6. Алгоритм Deep OC-SORT . . . . .	26
2.4. Выводы по главе . . . . .	26
<b>3 Разработка системы для проведения экспериментальных исследований . . . . .</b>	28
3.1. Выбор микрокомпьютера . . . . .	28
3.2. Выбор дополнительного вычислительного модуля . . . . .	29
3.3. Настройка и подключение компонентов . . . . .	29
3.4. Функциональная схема системы . . . . .	30
3.5. Выводы по главе . . . . .	30

<b>4 Проведение экспериментальной апробации . . . . .</b>	<b>31</b>
4.1. Исследование зависимости показателей качества от размера сети детектора . . . . .	31
4.2. Исследование зависимости показателей качества от размера ReID модели . . . . .	36
4.3. Исследование зависимости показателей качества от частоты кадров видеоизображения . . . . .	41
4.4. Исследование зависимости производительности от количества объектов на видеоизображении . . . . .	45
4.5. Выводы по главе . . . . .	49
<b>5 Методические указания на основе результатов экспериментов . .</b>	<b>51</b>
<b>ЗАКЛЮЧЕНИЕ . . . . .</b>	<b>52</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ . . . . .</b>	<b>53</b>

## **ВВЕДЕНИЕ**

Темпы роста количества собираемой информации стабильно увеличиваются. Системы анализа данных производят обработку всего этого потока. Они проводят миллионы вычислений, целью которых является выявление полезных закономерностей в повседневной жизни для повышения производительности труда и оптимизации расходов. Польза этого направления была признана руководителями из различных сфер: от компаний, ставящих целью максимизацию прибыли, до глав государственных институтов, улучшающих процессы взаимодействия миллионов людей.

Одним из наиболее бурно развивающихся направлений анализа данных является компьютерное зрение. За последние 20 лет прорыв в области глубоких нейронных сетей позволил решать задачи детектирования и классификации объектов на изображениях с принципиально недоступной до этого точностью. На основе новых возможностей происходит активный поиск способов отслеживания перемещений объектов интереса. По этой теме ежегодно выходит множество научных статей, применимость которых очевидна как в робототехнике, так и в бизнесе, урбанистике, архитектуре, государственном правлении и обеспечении порядка.

Эпоха интернета вещей проходила свое становление параллельно с развитием технологий обработки информации. Любой гаджет с подключением к интернету способен считывать сотни показаний ежедневно: начиная температурой воздуха и заканчивая поведенческими особенностями владельца. Весь этот поток сырых данных идет на централизованные сервера для последующей обработки, так как для этого требуются большие вычислительные мощности.

У описанного выше подхода есть свои минусы. Во-первых, увеличивается нагрузка на сеть, ведь необработанные данные, как правило, имеют больший вес. Во-вторых, возникают риски потери, вызванных техническими сбоями со стороны сложно устроенного сервера или проблемами с соединением. Перечисленные недостатки делают его вовсе не пригодным для некоторых задач, например, автономной робототехники, где непредвиденные задержки в сотые доли секунды могут привести к катастрофе.

Данная работа ставит своей целью исследовать проблему требовательных вычислений на микрокомпьютер для задачи отслеживания объектов на

видеоизображениях в городской среде. Исследование позволит оценить возможность проведения анализа сырых данных сразу в момент их получения и найти наиболее эффективные алгоритмы не только по метрикам качества, как это происходит в статьях, но и по скорости работы.

Для получения результата, соответствующего поставленной цели, составлен список задач, подлежащих решению в работе:

1. провести аналитический обзор существующих алгоритмов;
2. подготовить стенд для апробации;
3. подготовить эксперимент для сравнения различных алгоритмов на основании метрик НОТА, МОТА, IDF1 и показателей производительности.;
4. проанализировать полученные результаты.

# 1 Задача отслеживания объектов на видеоизображениях

Задача отслеживания объектов на видеоизображениях является одной из наиболее популярных среди исследователей в области компьютерного зрения. В этой главе будет представлено основывающееся на обзоре [1] краткое описание различных ее постановок, актуальности и проблем.

## 1.1. Постановка задачи

В общем случае задача подразумевает работу с видеоизображением с целью получения информации о объектах и разбивается на три цели: детектирование объектов в поле зрения, присвоение каждому из них уникального номера для идентификации и последующая запись истории перемещения. Способы их достижения разделяются на несколько основных подходов:

- МОТ (англ. Multiple Object Tracking – отслеживание нескольких объектов одновременно) или SOT (англ. Single Object Tracking – отслеживание одного объекта);
- работающие по мере поступления новых кадров (англ. online) или сразу со всем видео (англ. offline);
- использование двухкомпонентных (нейронная сеть детектор в сочетании с алгоритмом отслеживания, который работает с показаниями детектора) или однокомпонентных (единая сеть, которая сразу детектирует и отслеживает) систем;
- в случае двухкомпонентных систем: использующие ReID модели (англ. Re-identification – дополнительная нейронная сеть, задачей которой является исключительно создание векторов значений (англ. embedding), позволяющих отличить объекты одного класса друг от друга) или работающие без них.

В ходе работы будут рассмотрены только двухкомпонентные МОТ алгоритмы. Это связано с тем, что требования к вычислительным мощностям у них меньше, что является однозначным плюсом для запуска на маломощных устройствах. Более того, двухкомпонентные алгоритмы самые универсальные с точки зрения применения готовых решений – для адаптации под

новый набор данных достаточно переобучить сеть классификатор. Суммарность простоты использования и относительно низких вычислительных затрат выгодно выделяет этот класс алгоритмов и делает их идеальным выбором для множества областей, например, робототехники.

## 1.2. Основные проблемы при решении задачи

При решении поставленной задачи, возникает множество проблем. Среди них можно выделить следующие основные:

- пересечение объектов на изображении (например, люди в толпе), из-за чего системы испытывают трудности с сохранением присвоенных ранее уникальных номеров;
- компенсация движения камеры;
- изменения условий освещения, погоды или любых иных визуальных помехах;
- проблемы реидентификации из-за схожести объектов одного класса;
- проблемы реидентификации при временной потере объекта из поля зрения камеры;
- баланс между производительностью и качеством работы;
- работа в реальном времени и задержки обработки.

В ходе данной работы, основной фокус будет сделан на проблемах связанных с производительностью и задержками, а также запуске алгоритмов в реальном времени на малопроизводительных устройствах. Исследование данного вопроса поможет выработать общие рекомендации по выбору метода в зависимости от условий задачи (количество отслеживаемых объектов, качество видео и желаемая частота обновления), а также в целом проанализировать готовность технологии на данный момент к развертыванию на портативных устройствах.

### **1.3. Актуальность**

Методы отслеживания объектов на видеоизображениях находят свое применение в самых разнообразных областях. Удешевление стоимости вычислений позволяет их использовать для все большего количества задач, например:

- в спортивной аналитике, для более глубокого анализа аспектов игр, например, траекторий мяча (рисунок 1.1);

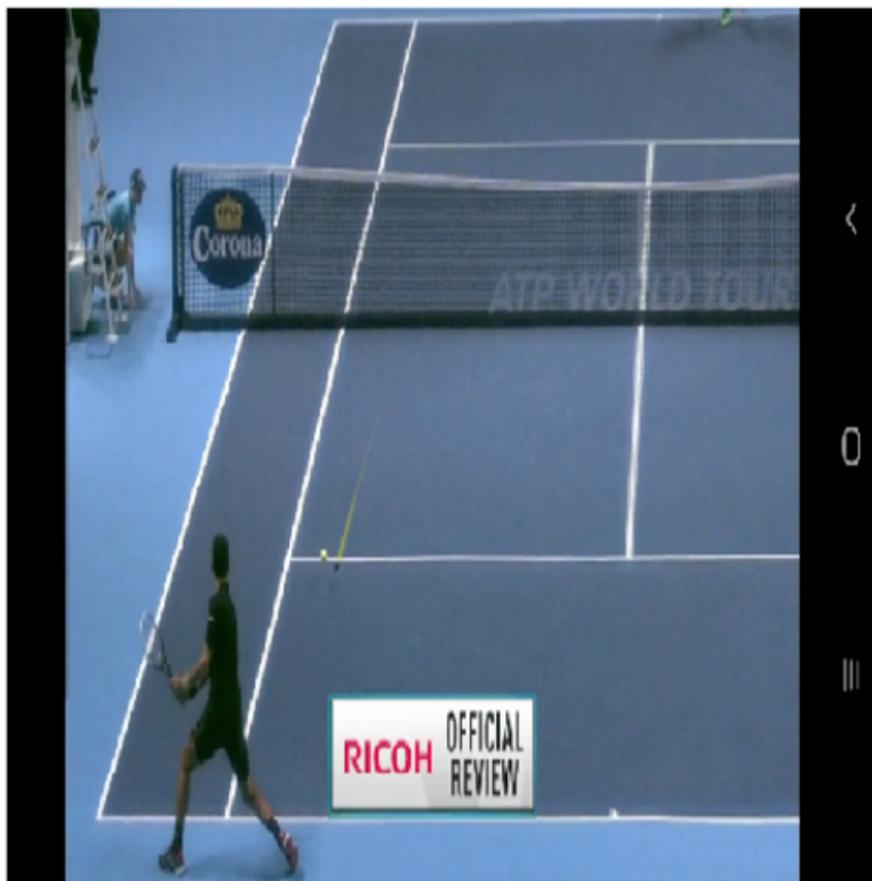


Рисунок 1.1 – Пример отслеживания мяча в игре теннис [2, страница 30, рисунок 16]

- в зоологии, для анализа поведения животных (рисунок 1.2);

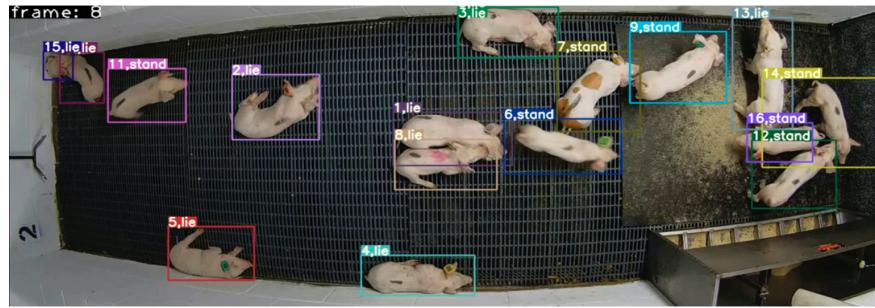


Рисунок 1.2 – Пример отслеживания поведения свиней [3, страница 12, рисунок 7]

- в урбанистике, для анализа паттернов передвижения или подсчета количества автомобилей (рисунок 1.3);

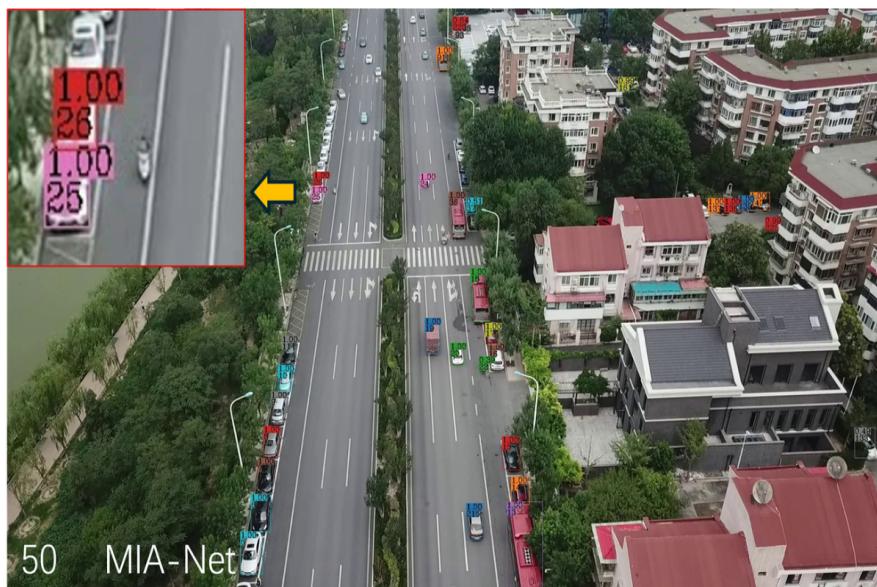


Рисунок 1.3 – Пример подсчета количества автомобилей с использованием дрона [4, страница 16, рисунок 8]

- в робототехнике, для отслеживания объекта (рисунок 1.3).

С увеличением доступности технологии естественным образом будет расти и количество способов использования. В свою очередь, это будет приводить к возрастанию производительности труда, развитию отраслей с возрастающей отдачей и экономики – повышению качества жизни. Именно поэтому данная работа ставит своей целью изучить возможности запуска методов отслеживания объектов на видеоизображениях на малопроизводительных устройствах.

## **1.4. Выводы по главе**

В первой главе была рассмотрена постановка задачи отслеживания объектов на видеоизображениях. Представлены основные современные парадигмы решения задачи, их основные отличия, а также выбрана наиболее перспективная для запуска на маломощных устройствах.

Показана актуальность технологии и важность исследования поставленной цели, решение которой потенциально полезно для множества отраслей экономики.

В следующих главах будут более подробно рассмотрены принципы работы современных методов отслеживания, рассмотрены метрики оценивания показателей качества, набор данных, на которых будет проводиться сравнение.

## **2 Обзор современных подходов к отслеживанию объектов на видеоизображениях**

В этой главе будут подробно рассмотрены следующие вещи:

- набор данных для сравнения различных алгоритмов друг с другом;
- различные метрики, по которым проводится сравнение;
- принцип работы и основные отличительные особенности выбранных алгоритмов.

### **2.1. Набор данных MOT17**

Набор данных MOTChallenge [5] является одним из наиболее популярных в области для сравнения между собой различных алгоритмов отслеживания нескольких объектов на видеоизображениях одновременно. Со дня своего появления он быстро обрел статус стандарта индустрии.

Бенчмарк (англ. Benchmark – тест производительности) MOT17 был выпущен в 2017 году и является идейным наследником более ранних версий – MOT15 и MOT16. В ходе этого развития авторами был исправлен ряд проблем старых версий:

- с развитием технологии отслеживания объектов на изображении MOT15 стал слишком простым для современных методов. MOT16 включает в себя ряд новых видео, где траектории объектов чаще пересекаются друг с другом, условия освещения более разнообразны, а средняя плотность объектов на изображении увеличилась в три раза;
- MOT17 решает другую важную проблему – уточняет разметку набора данных, так как в предыдущей версии нередко встречались ошибки.

В итоге MOT17 включается в себя 14 различных видео общей продолжительностью порядка 4 минут. Все они представляют собой различные записи с уличных камер, на которых пешеходы перемещаются по тротуарам, площадям, торговым центрам. В наборе данных представлены различные ракурсы: как видео с подвижной камерой снятые человеком, находящимся

в толпе, так и сделанные с возвышенности статично, например, камерой наблюдения.

На рисунках 2.1-2.3 приведены примеры изображений из бенчмарка.



Рисунок 2.1 – Пример изображения из набора данных MOT17 [5, страница 850, рисунок 3]



Рисунок 2.2 – Пример изображения из набора данных MOT17 [5, страница 850, рисунок 3]



Рисунок 2.3 – Пример изображения из набора данных МОТ17 [5, страница 850, рисунок 3]

## 2.2. Метрики для оценивания

В этом разделе разбираются основные метрики для оценивания качества отслеживания объектов на видеоизображениях.

### 2.2.1. Метрика МОТА

Метрика МОТА [6] (англ. Multiple Object Tracking Accuracy – точность отслеживания нескольких объектов) была предложена еще в 2008 году и является старейшей из всех используемых в работе. Ее авторы ставили целью создать критерий, который позволит оценивать точность выполнения задачи слежения. Подчеркивается важность способности метрики учитывать постоянство идентификации, точность оценки положения объекта и итоговой траектории. Более того, был сделан акцент на минимизации параметров, а также интуитивной понятности для человека и применимости для задач слежения в любой постановке.

В итоге был предложен следующий способ вычисления

$$\text{MOTA} = 1 - \frac{\sum_{t=0}^T (\text{FP} + \text{FN} + \text{IDSW})}{\sum_{t=0}^T g_t}, \quad (2.1)$$

где FP – количество неверно найденных объектов, которые на самом деле не были отмечены на изображении при разметке; FN – количество объектов, которые были отмечены при разметке, но не были найдены алгоритмом на изображении; IDSW – сколько найденных ранее объектов не были правиль-

но реидентифицированы и получили новый уникальный идентификационный номер,  $g_t$  – общее количество объектов, отмеченных при разметке на кадре. Из формулы 2.1 видно – чем больше показатель метрики, тем лучше справляется алгоритм.

Существует критика метрики МОТА, связанная с суммированием ошибок разного рода без какой-либо их предобработки, а также плохой репрезентацией при отслеживании с использованием нескольких камер. Тем не менее, МОТА используется практически в каждом исследовании по теме и на практике доказала свою состоятельность в оценивании качества отслеживания объектов на видеоизображениях. Именно поэтому было принято решение включить ее в качестве критерия для сравнения методов в данной работе.

### 2.2.2. Метрика IDF1

Метрика IDF1 [7] рекомендована для анализа производительности трекера в качестве дополнительной в связи с некоторым недостатками МОТА.

Авторы считали, что основная проблема других оценок – арифметические операции с разнородными ошибками, которые могут оказывать влияние на интерпретируемость полученного результата. В общем, они оценивали вероятность появления ошибок идентификации и детектирования в каждый момент времени, а не точность итоговых траекторий у каждого объекта.

Предложенная метрика IDF1 фокусировалась на оценке того, насколько качественно и долго трекер сохраняет идентичность объектов. Для этого, полученные с помощью трекера объекты, их идентичности и итоговые траектории сопоставляются с размеченными на основе методов графовой оптимизации, благодаря чему размеченные траектории сопоставляются с полученными на основе минимизации ошибки. Получившаяся оценка является более репрезентативной с точки зрения отслеживания конкретного объекта, так как оценивает именно задачу сохранения идентификации и точности итоговых траекторий. При этом у метрики есть и свои недостатки, поэтому авторы позиционируют ее как дополнительную для более комплексной оценки.

Метрики IDF1 представляет собой

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN}, \quad (2.2)$$

где IDTP – количество верно идентифицированных отслеживаемых объектов; IDFN – количество ненайденных объектов, которые были в разметке, но которым не были сопоставлены траектории; IDFP – количество ложных срабатываний, когда был найден несуществующий объект. Расчет всех этих значений происходит после упомянутого ранее процесса сопоставления полученных траекторий с размеченными на основе оптимизации графа.

В итоге, благодаря оценке с точки зрения качества сохранения присвоенных ранее идентификаций, метрика IDF1 является отличным дополнением для MOTA и была выбрана для учета в сравнении методов отслеживания объектов на видеоизображениях в рамках данной работы.

### 2.2.3. Метрика НОТА

НОТА [8] была представлена в 2020 году. Авторы поставили своей целью создать подход к оценке качества выполнения задачи отслеживания объектов на изображении, который будет оптимально сочетать в себе положительные черты таких метрик как MOTA, которая при оценке больше учитывает точность детектирования, так и IDF1, которая больше акцентирует внимание на точность идентификации.

Новая метрика, по задумке авторов, позволяет оценить насколько точно построенные алгоритмом отслеживания траектории сопоставляются с размеченными, при этом учитывая ошибки детектирования. Для этого используется вычисление на основе двух разделенных типов ошибок:

$$NOTA = \sqrt{DetA \times AssA}, \quad (2.3)$$

где DetA – точность детектирования объектов алгоритмом, а AssA – точность ассоциации этих объектов с итоговыми траекториями.

Для вычисления DetA используется классический подход для оценки

точности детектирования

$$\text{DetA} = \frac{\text{DetRe} \times \text{DetPr}}{\text{DetRe} + \text{DetPr} - \text{DetRe} \times \text{DetPr}}, \quad (2.4)$$

где:

$$\text{DetRe} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (2.5)$$

$$\text{DetPr} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (2.6)$$

Для AssA авторы вводят новый концепт

$$\text{AssA} = \frac{\text{AssRe} \times \text{AssPr}}{\text{AssRe} + \text{AssPr} - \text{AssRe} \times \text{AssPr}}, \quad (2.7)$$

где:

$$\text{AssRe} = \frac{1}{\text{TP}} \sum_{c \in TP} \frac{\text{TPA}(c)}{\text{TPA}(c) + \text{FNA}(c)}, \quad (2.8)$$

$$\text{AssPr} = \frac{1}{\text{TP}} \sum_{c \in TP} \frac{\text{TPA}(c)}{\text{TPA}(c) + \text{FPA}(c)}. \quad (2.9)$$

В формулах 2.8-2.9: TPA – количество правильно идентифицированных объектов; FPA – количество неправильно идентифицированных объектов, которых не было в разметке; FNA – количество неидентифицированных объектов, которые были в разметке.

Данный подход учитывает точность ассоциаций и выдает оценку ниже, если один объект за время работы отслеживался с различными идентификационными номерами, но его траектория была непрерывной на видео.

## 2.3. Обзор существующих алгоритмов

В данном разделе разбирается принцип работы выбранных для исследования алгоритмов.

### 2.3.1. Алгоритм ByteTrack

Алгоритм ByteTrack [9] был представлен в 2022 году и сразу же завоевал первые позиции по качеству отслеживания объектов на изображениях на основе метрик MOTA, HOTA и IDF1. До представления этого алгоритма,

методы в основном основывались на работе только на показаниях детектора с высокой степенью уверенностью. Авторы же предложили способ получить преимущество, используя так же показания, в которых детектор не очень уверен.

На вход алгоритма поступают:

- видеоизображения, на которых предстоит провести отслеживание объектов;
- детектор, который будет проводить детектирование объектов интереса;
- граничная степень уверенности детектора  $\tau$ .

Выходом алгоритма являются получившиеся траектории.

В ходе своей работы алгоритм сначала инициализирует массив траекторий для каждого найденного объекта. Затем в цикле для каждого кадра он получает набор пар область интереса-степень уверенности от детектора. Полученные пары разбиваются на две группы: с высокой степенью уверенности большей  $\tau$  и с низкой. После этого для ранее найденных траекторий с помощью фильтра Калмана строится догадка об их положении на новом кадре.

Теперь начинается первый этап сопоставления. Для каждого объекта из группы с высокой степенью уверенности алгоритм подбирает близкую траекторию на основе метрики похожести. В ее роли может выступать как IoU (англ. Intersection over Union – отношение площади пересечения к площади объединения двух областей), так и ReID модель.

На втором этапе сопоставления, для оставшихся траекторий, которым не нашлось объекта на первом этапе, производится попытка сопоставить объект из группы с низкой степенью уверенности. На этом этапе рекомендуется использовать IoU, так как зачастую такие объекты смазаны или имеют какие-то перекрытия с другими, а потому ReID модели дают ненадежный результат.

Оставшиеся траектории помещаются в буфер потерянных траекторий. Если они находятся в этом буфере больше, чем какое-то количество итераций алгоритма (как правило 30 кадров или 1 секунда), то они удаляются из

рассмотрения. Наличие этого буфера критически важно, так как часто объекты бывают перекрыты полностью на короткий срок, во время которого сопоставление их траектории невозможно. Для таких траекторий происходит интерполяция их положения.

В работе на наборе данных MOT17 при использовании изображений размером  $800 \times 1440$  (оригинальный в наборе данных  $1080 \times 1920$ ) получаются следующие показатели: MOTA – 80.3; IDF1 – 77.3; HOTA – 63.1.

Для обучения сети детектора они использовали не только MOT17, но и CrowdHuman [10] с ETHZ [11].

### 2.3.2. Алгоритм OC-SORT

Алгоритм OC-SORT [12] был представлен чуть позже ByteTrack в 2022 году. В своей работе авторы работают над проблемой интерполяции траекторий. Более ранние подходы использовали только фильтр Калмана и опирались исключительно на его оценки в случае перекрытия или потери объекта. Это приводит к очевидным минусам таким как:

- накопление шумов. Обычно скорость перемещения объектов на изображении всего несколько пикселей за кадр, а потому от кадра к кадру она может меняться в разы;
- в случае нелинейного движения линейная аппроксимация также приводит к большим ошибкам на продолжительных отрезках времени.

Эти недостатки не критичны в том случае, когда потеря случается на короткий промежуток времени. Однако, если использовать такую оценку на протяжении десятка кадров без обновления фильтра Калмана, то накопленная ошибка будет уже значительна и не позволит сопоставить вновь найденный объект с его изначальной траекторией. Авторы предлагают избежать этих проблем через смену подхода оценки на опирающийся больше на наблюдения.

Первая из них решается благодаря ORU (англ. Observation-Centric Re-Update – переобновление на основе наблюдений). После удачного сопоставления траектории с объектом после периода потери из поля зрения предлагается откатить состояние фильтра до того, что было в момент потери. После этого фильтр обновляют, подавая ему на вход виртуальную траекторию

от точки потери до точки нахождения. Благодаря этому удается нивелировать ошибку состояния фильтра, накопленную при его слепом обновлении во время потери.

Второй способ, использующийся в работе – ОСМ (англ. Observation-Centric Momentum – момент на основе наблюдений). На основе последних нескольких показаний (как правило 3), распределенных через равные, не слишком маленькие промежутки времени, строится предположение траектории движения. После этого находится точка, которая меньше всего от него отклоняется. Этот способ лучше традиционного использования последних двух точек и линейной экстраполяции, так как меньше подвержен шумам.

Полученный подход работает чуть лучше, чем ByteTrack: MOTA – 78.0; IDF1 – 77.5; HOTA – 63.2.

### 2.3.3. Алгоритм BoT-SORT

BoT-SORT [13] третий алгоритм, представленный в 2022 году. В своей работе авторы пишут, что алгоритмы, представленные ранее имеют множество недостатков. Во-первых, фильтр Калмана используется для предсказания отношения ширины области интереса к ее высоте, что является причиной неточности оценки. Во-вторых, авторы утверждают, что в случае временной потери объекта из поля зрения реидентификация зачастую проваливается из-за неучтенного движения камеры. В-третьих, использование ReID моделей ведет к оптимизации метрики IDF1, в то время как IoU оптимизирует MOTA, поэтому существует вероятность улучшения производительности по обеим метрикам через введение нового способа определения похожести на основе двух этих.

Для решения первой проблемы было предложено изменить классический для задачи отслеживания объектов на изображении вектор состояния фильтра Калмана

$$x = [x_c, y_c, a, h, \dot{x}_c, \dot{y}_c, \dot{a}, \dot{h}]^T, \quad (2.10)$$

где  $(x_c, y_c)$  – координаты области интереса,  $a$  – отношение сторон области интереса, а  $h$  – ее высота. Вместо этого предложен вектор состояния

$$x = [x_c, y_c, w, h, \dot{x}_c, \dot{y}_c, \dot{w}, \dot{h}]^T, \quad (2.11)$$

где  $w$  – ширина области интереса. Авторы не очень понимают, почему это улучшает результат, но метрики растут.

Для решения второй они добавляют использование алгоритма СМС (англ. Camera Motion Compensation – компенсация движения камеры). При допущении, что объекты от кадра к кадру перемещаются по изображению не сильно, данный подход значительно повышает робастность сопоставления траекторий и найденных объектов при нестабильном положении камеры в пространстве.

Последняя проблема решается через комбинацию матриц близости IoU и косинусовых расстояний близости ReID модели. Для этого просто берется минимум для каждой пары объектов.

Применение всех перечисленных подходов позволило добиться лучшего результата среди всех трекеров на тот момент: MOTA – 78.5; IDF1 – 82.1; HOTA – 69.2.

#### 2.3.4. Алгоритм StrongSORT

Алгоритм StrongSORT [14] был представлен в 2023 году. Авторы поднимают вопрос отсутствия стандарта тестирования различных алгоритмов, в связи с чем иногда преимущество получают не те, которые работают эффективнее, а для которых был лучше обучен детектор. Это затрудняет процесс объективного сравнения современных подходов между собой. Также авторы утверждают, что если использовать старый алгоритм DeepSORT [15], но дооснастить его современными решениями, то получится конкурентоспособный подход. Свое желание актуализировать DeepSORT они мотивируют его легковесностью и эффективностью.

В итоге к алгоритму DeepSORT были добавлены:

- улучшенный экстрактор особенностей объектов;
- компенсация движения камеры;
- модифицированный для лучшей работы с шумами фильтр Калмана;
- обновление особенностей объекта с помощью скользящего экспоненциального среднего;

- реидентификация не только по внешнему виду, но и по расстоянию Махalanобиса.

Более того, авторы добавляют два новых алгоритма AFLink и GSI, которые позволяют качественнее обрабатывать моменты, когда объект теряется из кадра или перекрывается другими.

Работа авторов демонстрирует эффективность модульного подхода в задаче отслеживания объектов на изображении. Алгоритм StrongSORT был получен наследованием друг на друга различных подходов, а также добавлением нескольких новых методик. Каждая из модификаций дает прирост производительности в несколько пунктов, что в общей сумме повышает эффективность алгоритма и позволяет ему обойти все современные по показателям метрик: MOTA – 77.1; IDF1 – 82.3; HOTA – 69.6.

### 2.3.5. Алгоритм ImprAssOC

Алгоритм ImprAssOC [16] также был представлен в 2023 году.

Более ранние алгоритмы, как это было упомянуто в описании метода ByteTrack, делили объекты на две группы: с низкой и высокой степенью уверенности. Из-за этого объекты из группы с низкой степенью уверенности не могли быть сопоставлены с треками, объект которых потерян, а также в целом приоритет всегда отдавался объектам с высокой. Авторы избегают этого, составляя матрицу расстояний сразу для всех объектов. Расстояния до объектов из группы с низкой степенью уверенности после этого нормализуются, чтобы быть одной размерности.

Более того, авторы улучшают алгоритм ассоциации, потому что посчитали, что используемое в BoT-SORT взятие минимума из IoU и ReID не использует все доступные данные, а StrongSORT использует расстояние Махalanобиса, которое по их мнению дает только грубую оценку при высокой степени неизвестности. Для этого они комбинируют подходы и берут взвешенную сумму визуальных особенностей и расстояния до траектории, в случае если IoU больше какого-то значения, иначе просто присваивается максимальная степень непохожести.

Также авторы вводят новый подход для инициализации траектории. Раньше это либо происходило сразу для любого найденного объекта, что

провацировало создание лишних траекторий для FP результатов детектирования, либо инициализировали траекторию, если объект был найден n раз подряд, что искусственно создавало кадры, на которых объект найден, но еще не отслеживается. Суть подхода заключается в том, чтобы откинуть все FP объекты. Для всех несопоставленных с траекторией объектов вычисляется IoU с сопоставленными. Если больше какого-то граничного значения, то результат детектирования признается дубликатом и отбрасывается.

Для обучения сети детектора используется MOT17, MOT20, CrowdHuman, ETH и CityPersons [17]. Получены следующие показатели: MOTA – 82.2; IDF1 – 82.1; HOTA – 66.4.

### 2.3.6. Алгоритм Deep OC-SORT

Алгоритм Deep OC-SORT [18] последний из рассматриваемых в этой работе. Авторы решили совместить Deep-SORT и OC-Sort. Это чем-то похоже на StrongSORT, но со своими особенностями.

Во-первых, предложен улучшенный алгоритм компенсации движения камеры. Так, фильтр Калмана, ОСМ и ORU получают на вход данные с компенсацией погрешностей, вызванных ее движением.

Во-вторых, предложен улучшенный алгоритм скользящего экспоненциального среднего. Авторы по показателю уверенности в результате детектирования адаптивно меняют параметр, отвечающий за быстроту усреднения. Мотивация этого кроется в том, что если степень уверенности низкая, объект либо смазан, либо перекрыт другими объектами, а значит делать поправку на новые особенности стоит в меньшей степени.

В итоге полученный подход работает относительно лучше, чем OC-Sort: MOTA – 75.6; IDF1 – 79.2; HOTA – 63.9. Хотя MOTA немного и упала, но IDF1 и HOTA выросли, что свидетельствует о лучшей работе с реидентификацией и особенностями объектов.

## 2.4. Выводы по главе

Во второй главе было рассмотрены:

- набор данных MOT17, на котором будет проводиться сравнение;

- метрики, принцип вычисления каждой и концептуальные отличия полученных оценок;
- разобран общий принцип работы методов отслеживания объектов на видеоизображениях, сравнение которых будет проведено в последующих главах.

### **3 Разработка системы для проведения экспериментальных исследований**

В данной главе будут рассмотрены выбранные для системы апробирования компоненты и описан процесс их подготовки.

#### **3.1. Выбор микрокомпьютера**

На данный момент на рынке представлено множество микрокомпьютеров. Наиболее распространенными из них являются Orange Pi, Nvidia Jetson и Raspberry Pi.

Идеальным выбором для запуска тяжеловесных алгоритмов является Nvidia Jetson, но он обладает рядом недостатков:

- высокая цена;
- высокое энергопотребление;
- относительно низкая распространенность в России;
- большой вес у старших моделей.

Перечисленные недостатки критичны, особенно в задачах мобильной робототехники, где энергоэффективность и рабочий вес наиболее чувствительные параметры. Orange Pi является подобием Raspberry Pi, но при этом обладает худшой поддержкой и меньшей распространенностью.

Raspberry Pi является идеальным выбором в силу следующих причин:

- низкая цена – в России можно найти предложения за 10000 рублей;
- малый вес – плата весит 50г;
- высокая энергоэффективность;
- доступность в России;
- отличная поддержка;
- большое количество информации в интернете;

- внешние вычислительные модули TPU, разработанные специально для платформы.

Именно поэтому для стенда апробации был выбран микрокомпьютер Raspberry PI 5 8GB.

### **3.2. Выбор дополнительного вычислительного модуля**

В качестве дополнительного вычислительного модуля в работе будет использоваться TPU (англ. Tensor Processing Unit – устройство обработки тензоров). Использование TPU позволяет оптимизировать работу с тензорами с помощью параллельных вычислений. Благодаря этому, скорость работы нейронных сетей возрастает в разы. На данный момент на рынке РФ представлены TPU двух компаний: Google Coral и Hailo.

Hailo является более новой разработкой, мощнее и эффективнее. Однако, в связи с его новизной, все еще остаются проблемы и баги, которые могут быть критичны при использовании. Так же сам процесс взаимодействия с Hailo требует переписывания всего кода, так как требует использования отдельной библиотеки.

В свою очередь Google Coral требует только конвертировать веса нейронной сети в специальный формат, после чего все происходит в автоматическом режиме. Для его работы не требуется переписывать ни единой строчки кода. Также существуют различные версии, которые позволяют подключение с использованием различных портов: M2, Mini PCIe, USB. Существует еще и версия с двумя ядрами, что дает возможность одновременно запускать две разных нейронных сети независимо друг от друга.

По совокупности удобства использования, доступности и проверенной временем надежности выбор пал на внешний вычислительный модуль Google Coral M.2 Accelerator with Dual Edge TPU и плата для подключения Pineboards AI Dual.

### **3.3. Настройка и подключение компонентов**

Для удобства работы была выбрана операционная система Raspberry PI OS, так как она сразу включает в себя все нужные драйвера для Raspberry

Pi, а также удобный интерфейс для их настройки. Дополнительно нужно только поставить драйвера Google Coral, представленные на официальном сайте Google Coral.

### 3.4. Функциональная схема системы

Итоговая функциональная схема системы представлена на рисунке 3.1. При тестировании алгоритмов на вход микрокомпьютера будет пода-

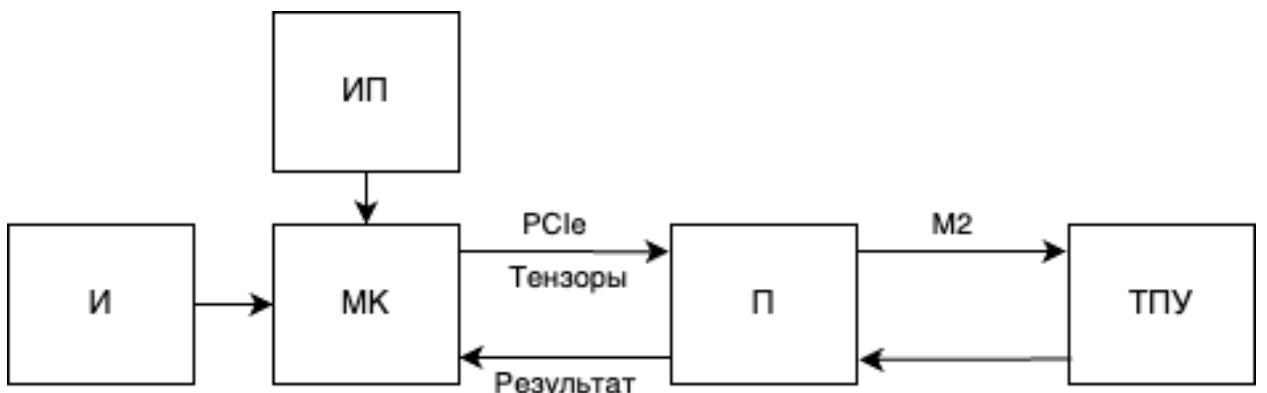


Рисунок 3.1 – Функциональная схема системы: И – изображение; ИП – источник питания; МК – микрокомпьютер Raspberry PI 5; П – плата Pineboards AI Dual; ТПУ – Google Coral TPU Dual;

ваться изображение. После первичной обработки оно посредством платы передается на ТПУ для проведения вычисления нейронной сети детектора, после чего результат возвращается обратно на микрокомпьютер для последующей обработки алгоритмом отслеживания.

### 3.5. Выводы по главе

В главе были разобраны представленные на рынке варианты микрокомпьютеров и вычислительных модулей ТПУ. В итоге выбраны комплектующие для стенда апробации:

- микрокомпьютер – Raspberry Pi 5 8GB.
- вычислительный модуль – Google Coral TPU Dual;
- плата для подключения – Pineboards AI Dual.

Общая стоимость компонент составляет 28000 рублей; масса – 100г.

## **4 Проведение экспериментальной апробации**

В этой главе для всестороннего анализа качества работы алгоритмов на основе показаний метрик MOTA, HOTA и IDF1 и производительности будут описаны и проведены следующие эксперименты:

- исследование влияния размера сети детектора;
- исследование влияния различных ReID моделей;
- исследование влияния частоты кадров видеоизображения;
- исследование влияния количества отслеживаемых объектов на производительность.

### **4.1. Исследование зависимости показателей качества от размера сети детектора**

Первым экспериментом было решено провести сравнение показателей метрик HOTA, MOTA и IDF1 от размера сети детектора на двух различных размерах изображений. Это исследование поможет определить, насколько значителен вклад увеличения сети детектора и размера изображения.

Для проведения экспериментов было произведено вычисление показателей качества на наборе данных MOT17 с различными конфигурациями: размер изображения 320 или 512; детекторы в возрастающем порядке – yolov8n, yolov8s, yolov8m, yolov8l, yolov8x. Результаты исследования представлены на рисунках 4.1-4.6. Как и ожидалось, чем больше сеть, тем лучшее качество работы по каждой из метрик она дает, не считая мелких отклонений.

По графикам видно, что в целом самая маленькая модель yolov8n на размере изображения 512 на 512 работает по качеству сопоставимо с самой большой yolov8x на размере изображений 320 на 320. При этом, прирост производительности для большой сети примерно в два раза меньше, чем для маленькой. В связи с этим становится очевидно, что в условиях ограниченных вычислительных мощностей, следует делать выбор в пользу увеличения разрешения видеоизображения, а не более мощного детектора.

Точные значения результата сравнения между собой yolov8n и yolov8x на разных разрешениях видеоизображения представлены в таблицах 4.1 - 4.3.

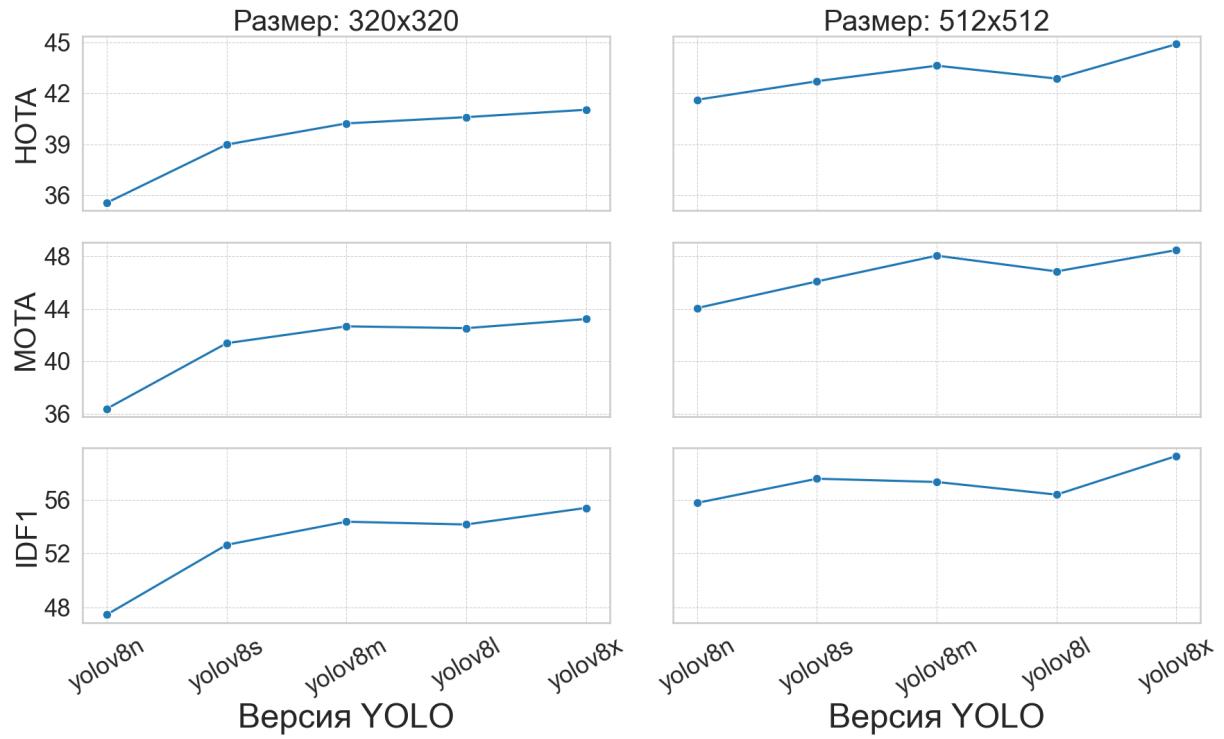


Рисунок 4.1 – График зависимости метрик HOTA, MOTA и IDF1 от размера сети детектора для алгоритма ByteTrack

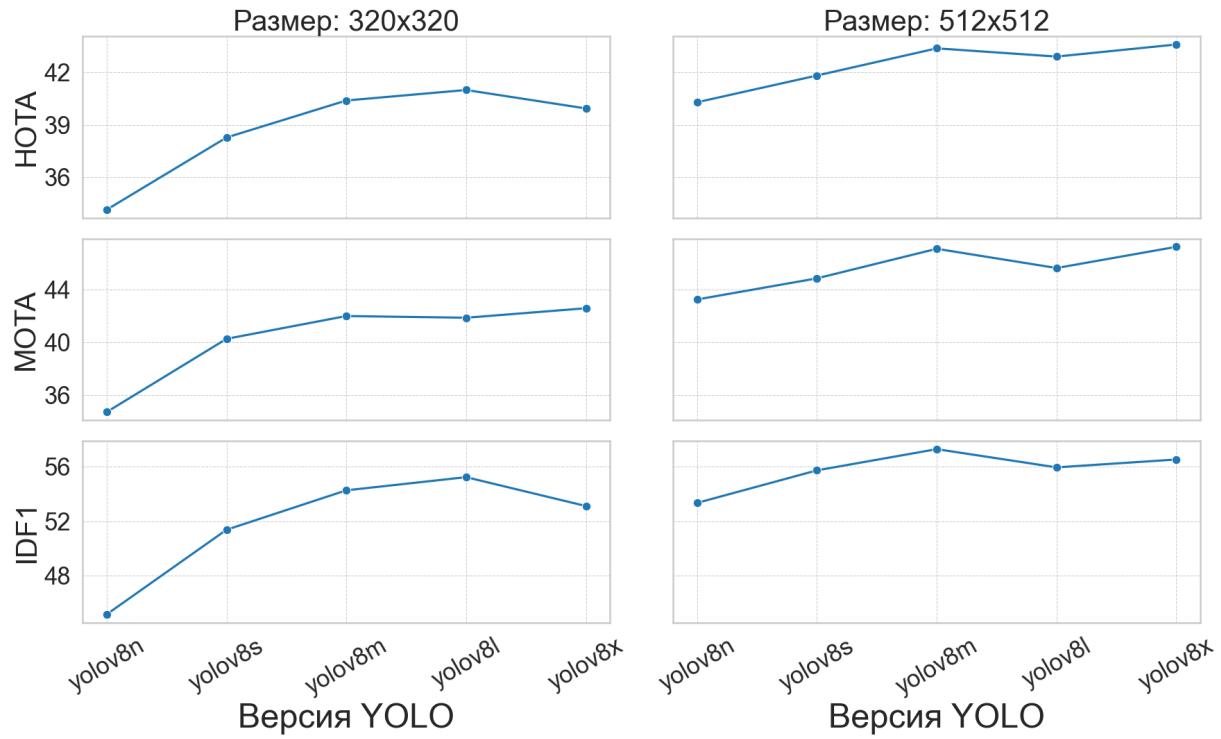


Рисунок 4.2 – График зависимости метрик HOTA, MOTA и IDF1 от размера сети детектора для алгоритма OC-SORT

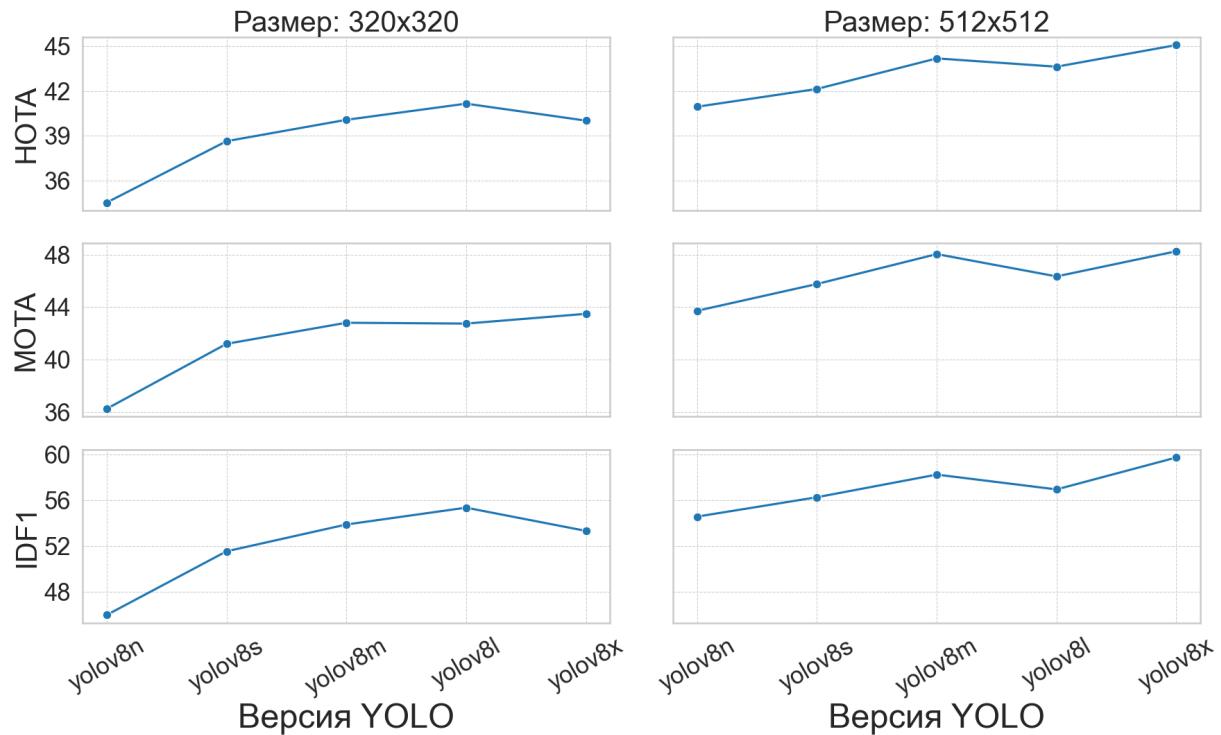


Рисунок 4.3 – График зависимости метрик HOTA, MOTA и IDF1 от размера сети детектора для алгоритма Deep OC-SORT

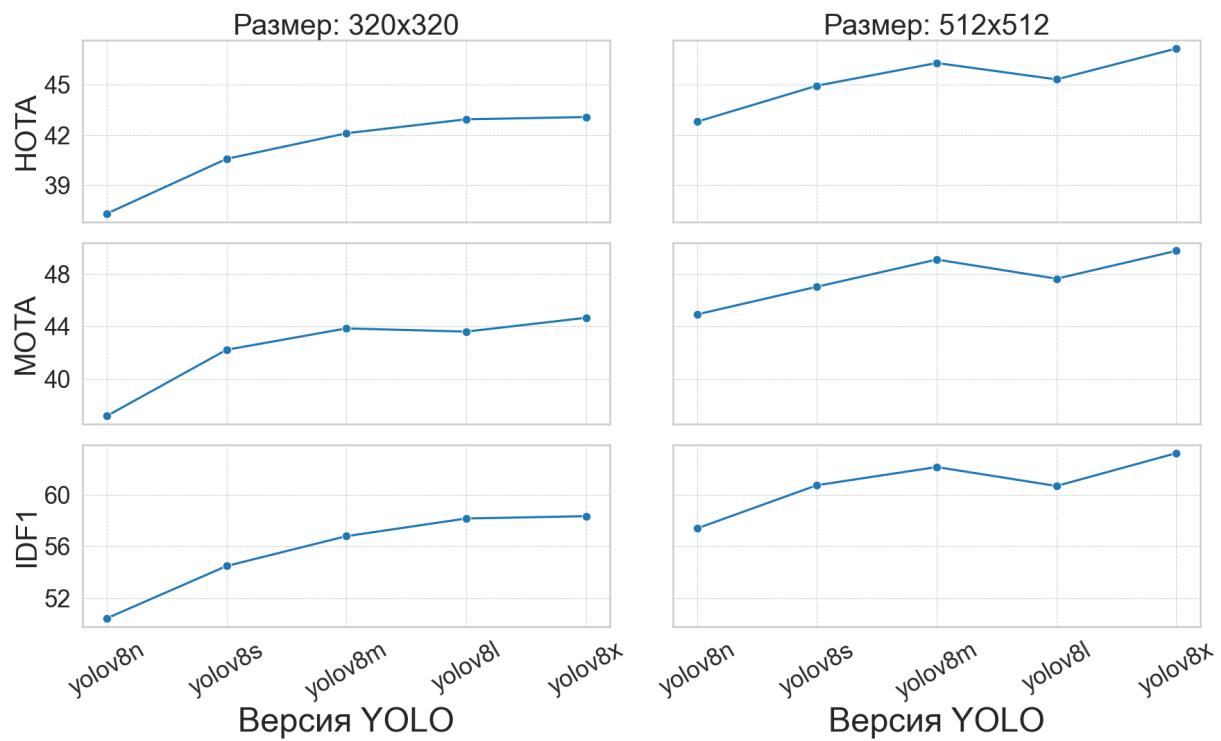


Рисунок 4.4 – График зависимости метрик HOTA, MOTA и IDF1 от размера сети детектора для алгоритма StrongSORT

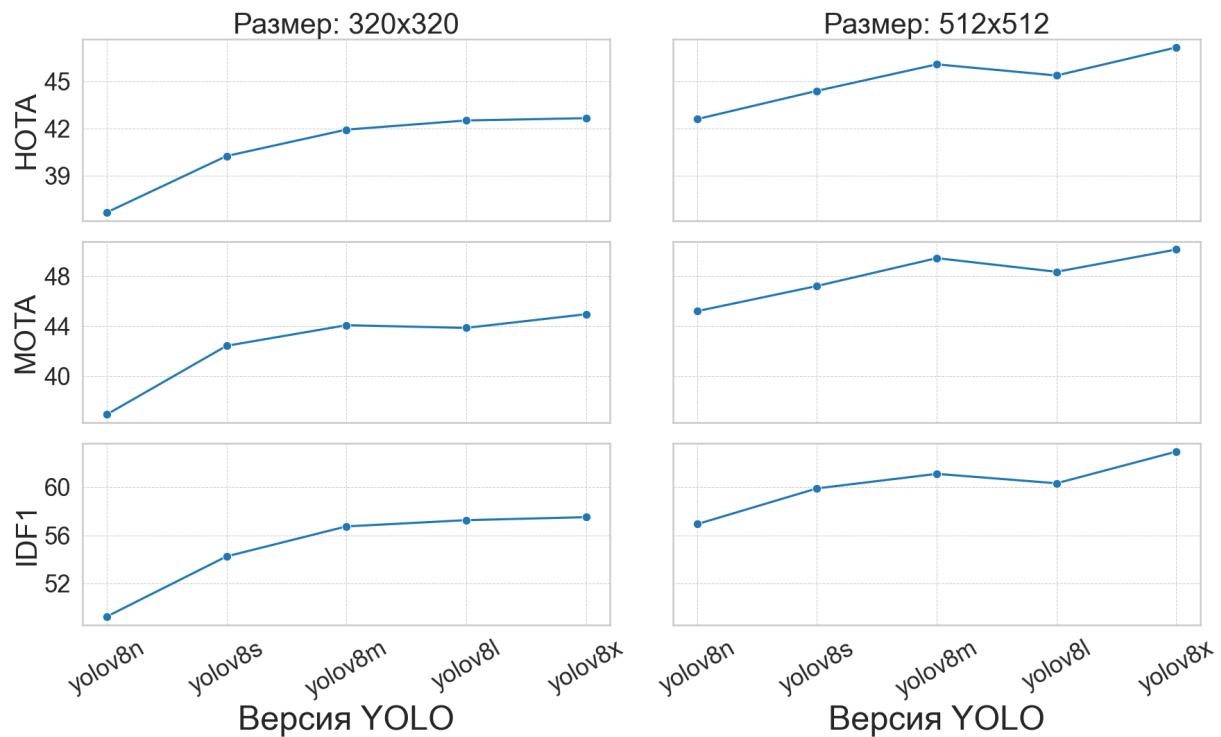


Рисунок 4.5 – График зависимости метрик HOTA, MOTA и IDF1 от размера сети детектора для алгоритма BoT-SORT

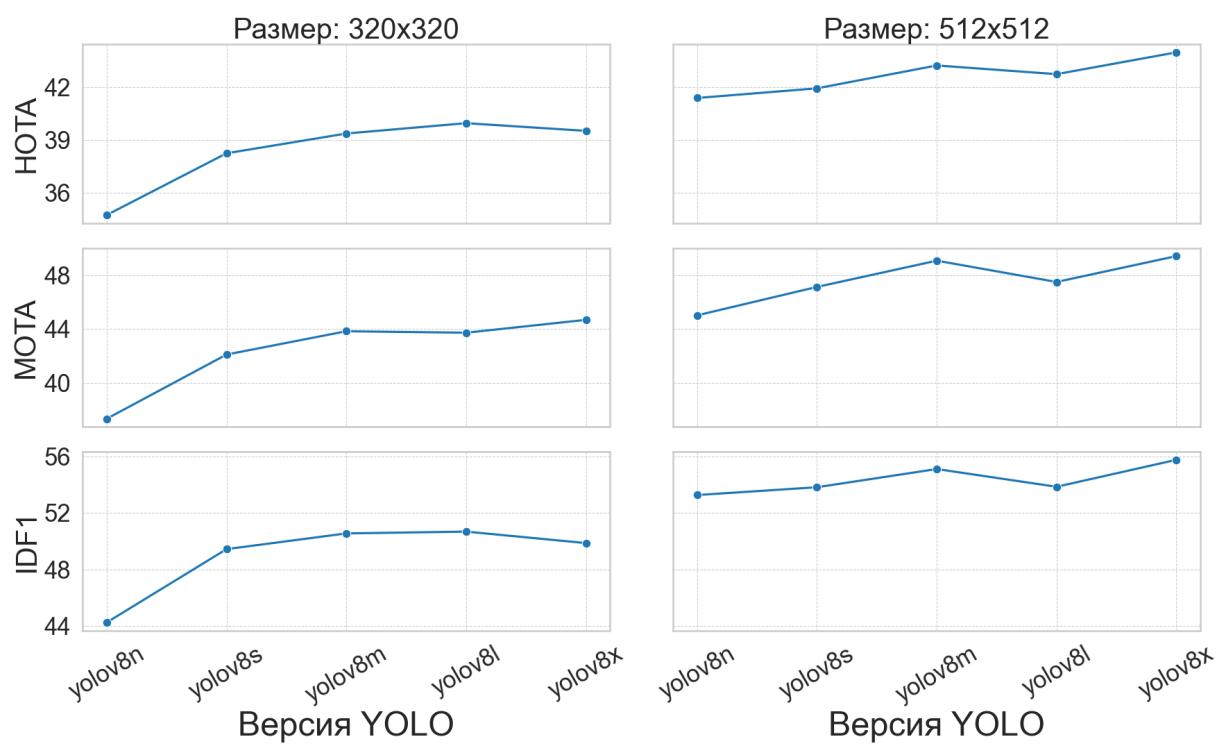


Рисунок 4.6 – График зависимости метрик HOTA, MOTA и IDF1 от размера сети детектора для алгоритма ImprAssOC

Таблица 4.1 – Среднее значение метрики НОТА для yolov8n и yolov8x

	320	512	yolov8n	yolov8x
	yolov8n	yolov8x	yolov8n	yolov8x
BoT-SORT	32.8	38.1	37.6	41.7
ByteTrack	31.0	35.2	35.2	38.2
Deep OC-SORT	30.6	35.1	35.4	38.9
ImprAssOC	34.0	38.3	38.8	42.0
OC-SORT	28.8	33.2	33.7	37.0
StrongSORT	34.5	39.3	38.9	42.7

Таблица 4.2 – Среднее значение метрики МОТА для yolov8n и yolov8x

	320	512	yolov8n	yolov8x
	yolov8n	yolov8x	yolov8n	yolov8x
BoT-SORT	33.5	41.4	40.6	45.8
ByteTrack	32.2	38.2	37.5	41.2
Deep OC-SORT	31.1	37.9	37.6	42.0
ImprAssOC	36.2	44.1	43.3	48.5
OC-SORT	28.0	34.9	34.6	38.5
StrongSORT	34.0	41.1	40.5	45.5

Таблица 4.3 – Среднее значение метрики IDF1 для yolov8n и yolov8x

	320	512	yolov8n	yolov8x
	yolov8n	yolov8x	yolov8n	yolov8x
BoT-SORT	43.7	50.6	50.0	55.4
ByteTrack	40.9	46.3	45.9	49.8
Deep OC-SORT	40.7	46.5	46.9	51.7
ImprAssOC	44.3	48.9	50.2	53.7
OC-SORT	37.1	43.1	43.5	47.6
StrongSORT	47.0	53.3	52.6	57.6

## 4.2. Исследование зависимости показателей качества от размера ReID модели

Вторым экспериментом было проведение сравнения результатов для различных ReID моделей. Благодаря этому, можно будет понять, насколько использование более требовательных моделей оправдано.

В ходе эксперимента, для каждой сети детектора и размера изображения из прошлого пункта был проведен анализ точности работы отдельных ReID моделей на наборе данных MOT17. Полученные результаты можно увидеть на рисунках 4.7-4.12. Отдельно стоит отметить, что на рисунках 4.7-4.8 нет влияния выбора, так как реализации алгоритмов ByteTrack и OC-SORT не используют ReID модели, а основываются на метрике IoU.

Полученные графики явно показывают, что только самая маленькая ReID модель OSNet x0.25 дает стабильно более низкий результат. При этом все остальные модели дают абсолютно схожие показатели с минимальными улучшениями относительно друг друга.

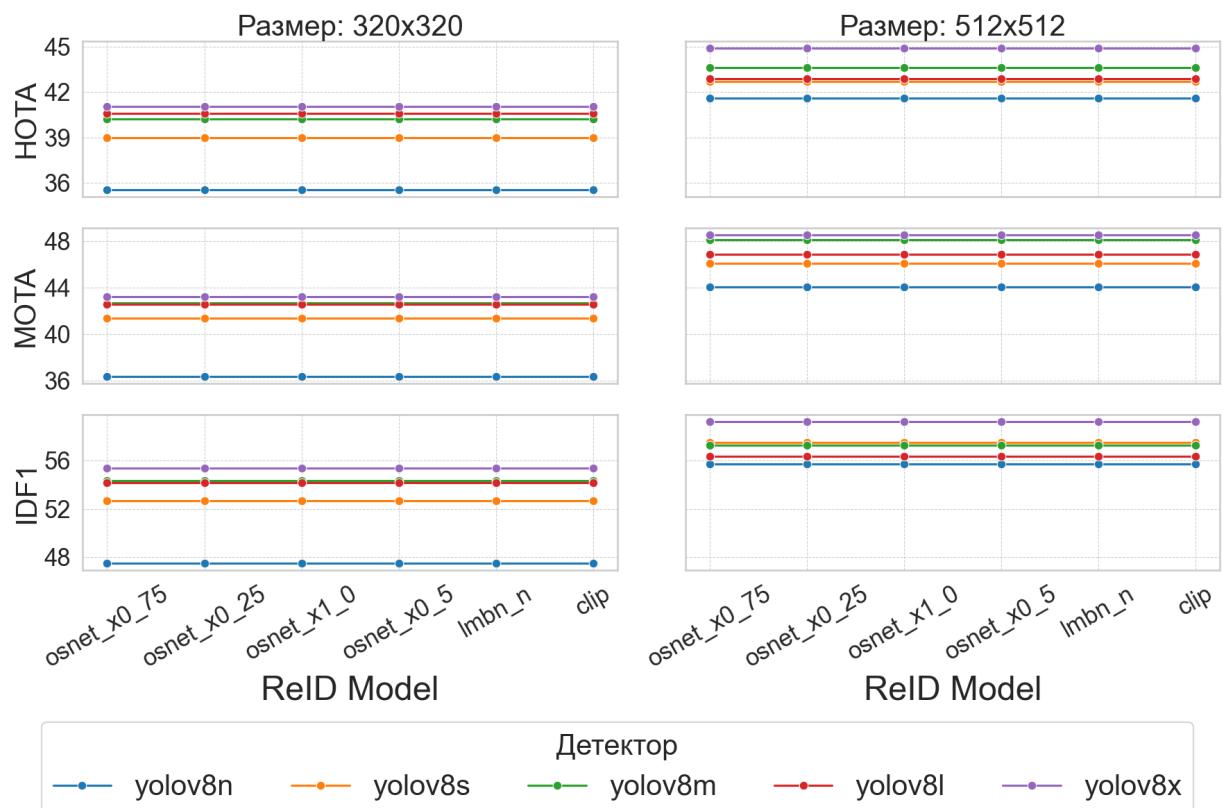


Рисунок 4.7 – График зависимости метрик HOTA, MOTA и IDF1 от ReID модели для алгоритма ByteTrack

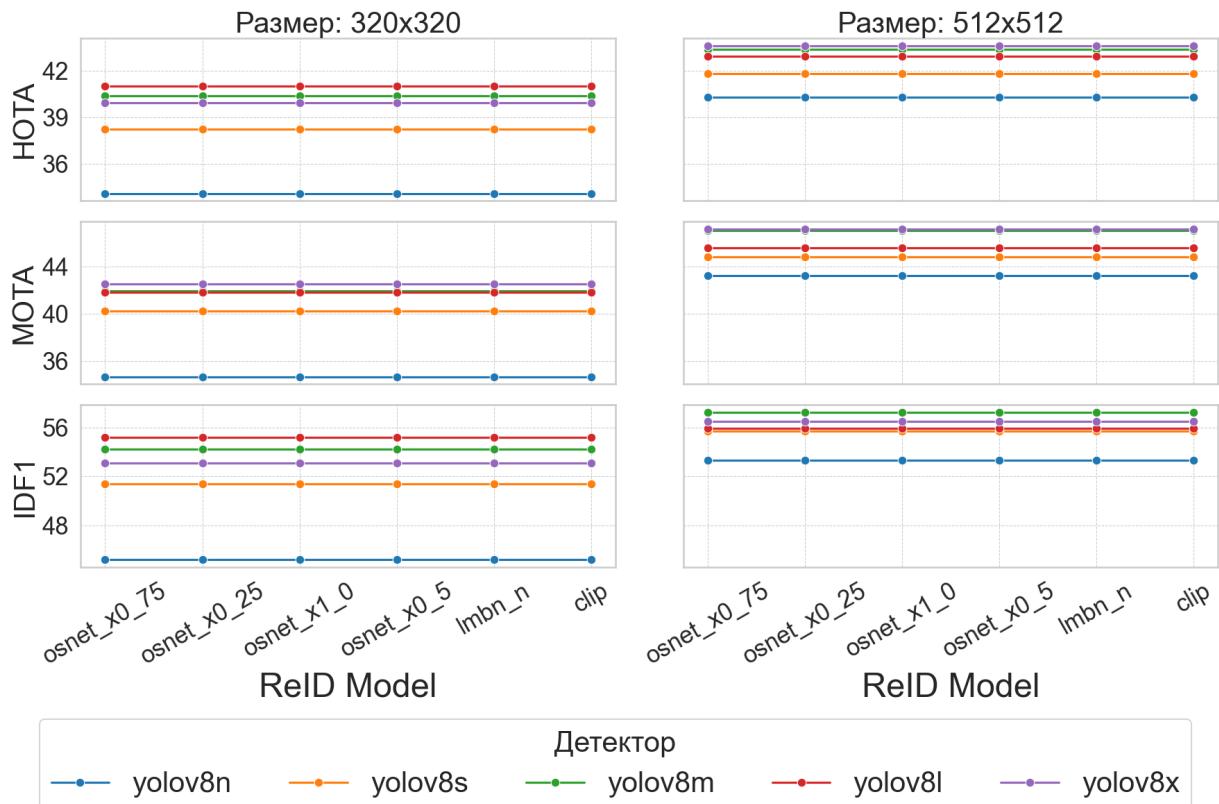


Рисунок 4.8 – График зависимости метрик HOTA, MOTA и IDF1 от ReID модели для алгоритма OC-SORT

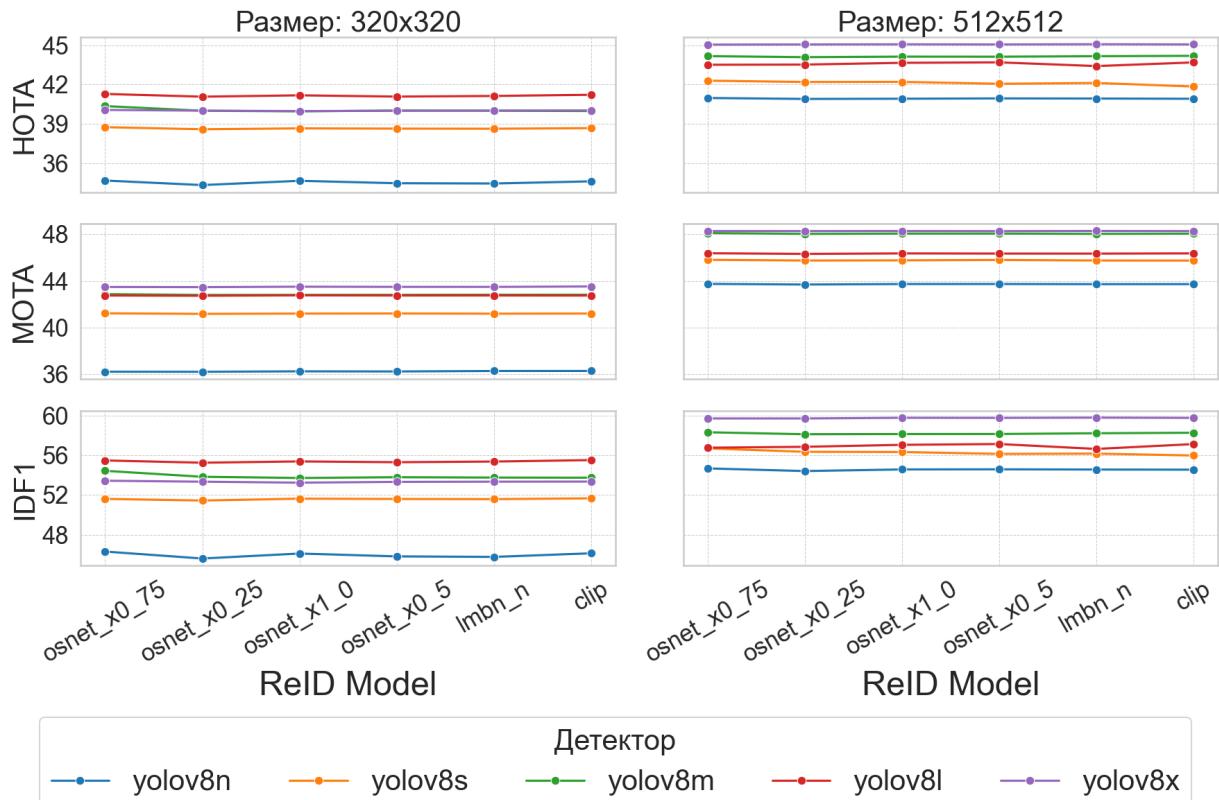


Рисунок 4.9 – График зависимости метрик HOTA, MOTA и IDF1 от ReID модели для алгоритма Deep OC-SORT

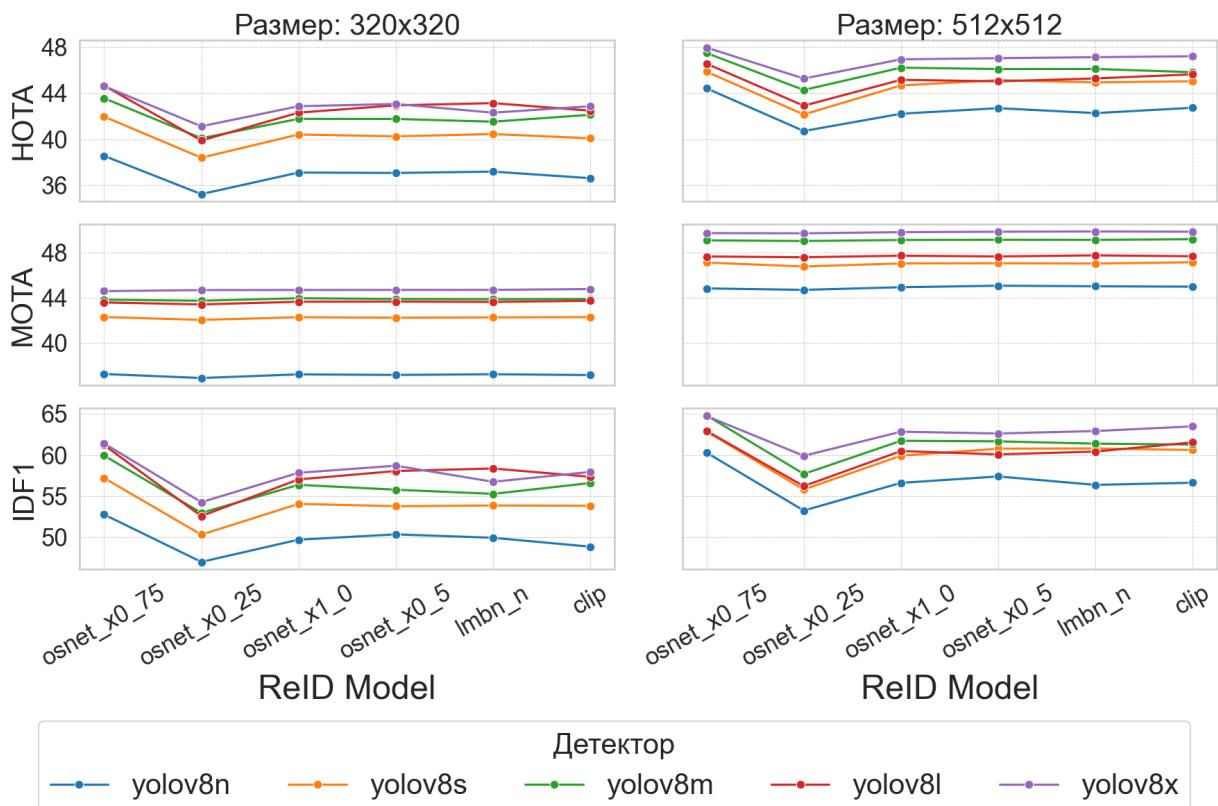


Рисунок 4.10 – График зависимости метрик НОТА, МОТА и IDF1 от ReID модели для алгоритма StrongSORT

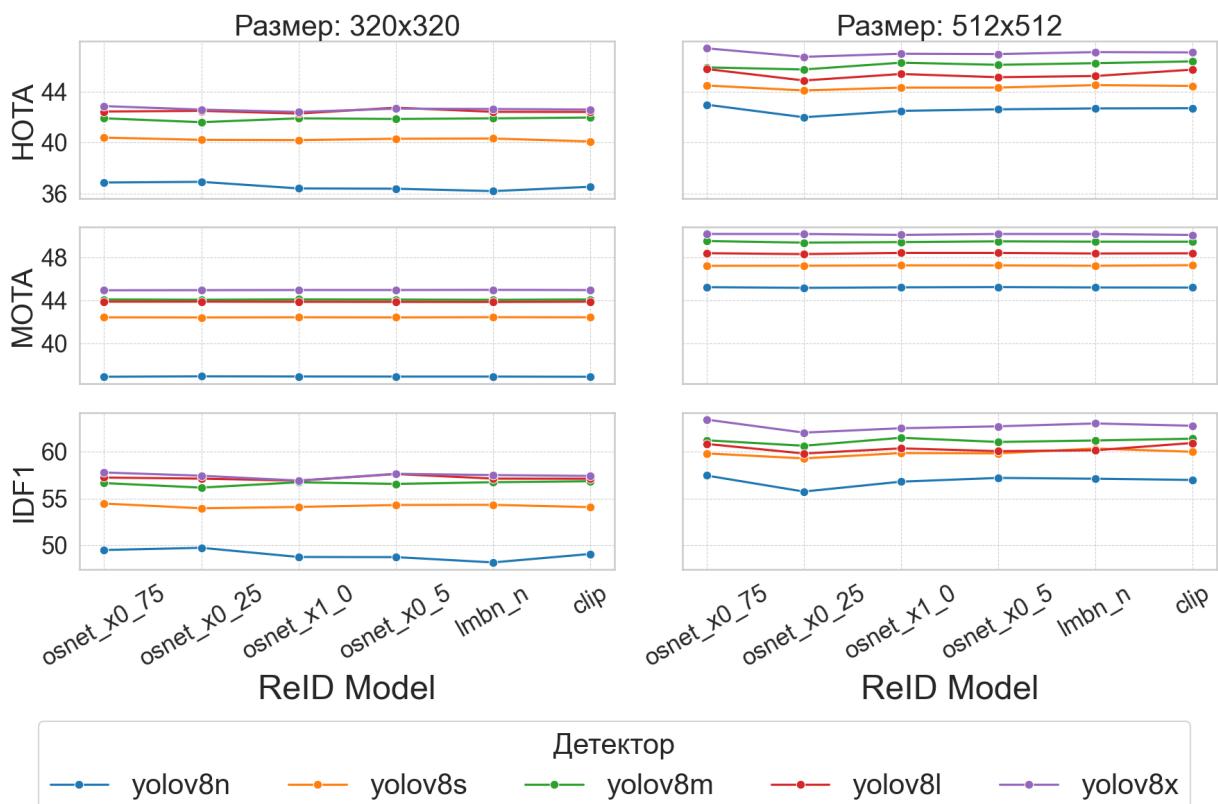


Рисунок 4.11 – График зависимости метрик НОТА, МОТА и IDF1 от ReID модели для алгоритма BoT-SORT

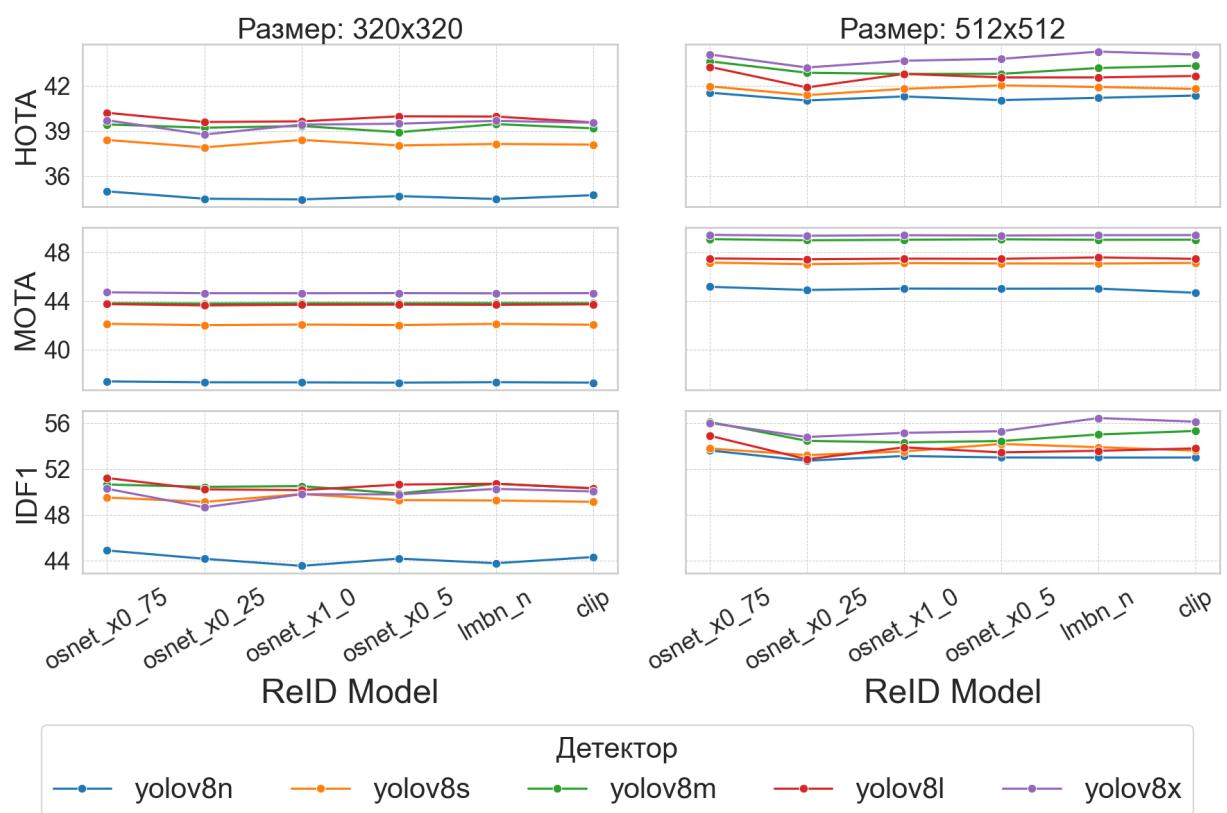


Рисунок 4.12 – График зависимости метрик HOTA, MOTA и IDF1 от ReID модели для алгоритма ImprAssOC

## **4.3. Исследование зависимости показателей качества от частоты кадров видеоизображения**

В ходе работы также было проведено исследование зависимости показателей качества от частоты кадров видеоизображения. Мотивация этого сравнения заключается в том, что при запуске на микрокомпьютере скорость работы заметно ниже, а значит временные промежутки между кадрами больше. В следствии этого, объекты на изображении будут перемещаться сильнее между кадрами, что повлияет на качество отслеживания. Именно поэтому, для адекватного сравнения алгоритмов между собой, важно пересчитать метрики для разных показателей частоты кадров.

На рисунках 4.13-4.18 виден результат проведенного сравнения. Все показатели медианные для каждого алгоритма. Для анализа выбраны детекторы yolov8n и yolov8s, так как только они способны запускаться на микрокомпьютере с частотой кадров больше 10, а также yolov8x как эталон возможного максимума. Частоты кадров видеоизображения для сравнения подобраны методом кластеризации k-средних всех полученных результатов частоты работы на микрокомпьютере. По результатам эксперимента получена таблица 4.4.

Таблица 4.4 – Корреляция метрики НОТА с частотой кадров видеоизображения

Частота кадров, Гц	(0, 6]	[6, 30]
BoT-SORT	0.91	0.21
ByteTrack	0.93	0.44
Deep OC-SORT	0.91	0.36
ImprAssOC	0.71	0.36
OC-SORT	0.92	0.48
StrongSORT	0.89	0.16

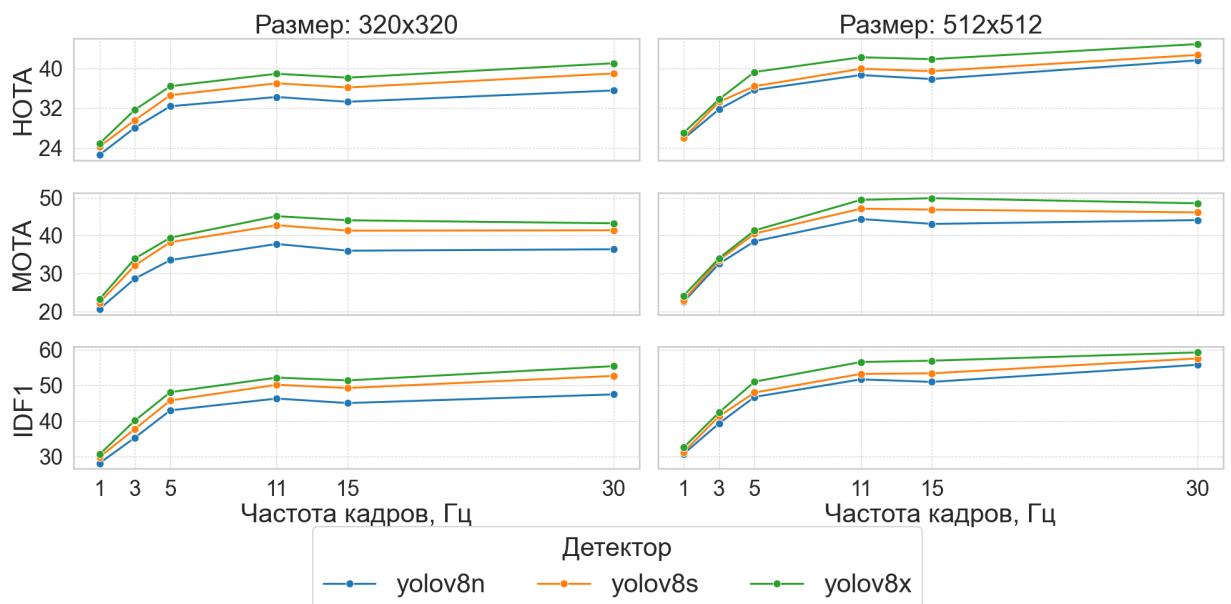


Рисунок 4.13 – График зависимости метрик НОТА, МОТА и IDF1 от частоты кадров видеоизображения для алгоритма ByteTrack

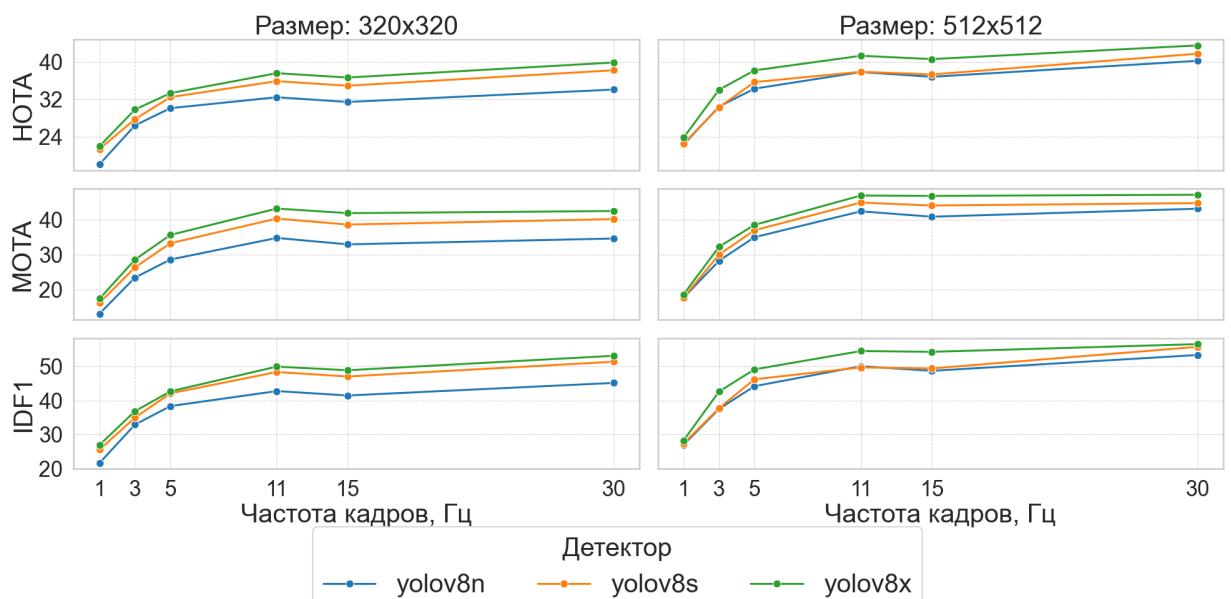


Рисунок 4.14 – График зависимости метрик НОТА, МОТА и IDF1 от частоты кадров видеоизображения для алгоритма OC-SORT

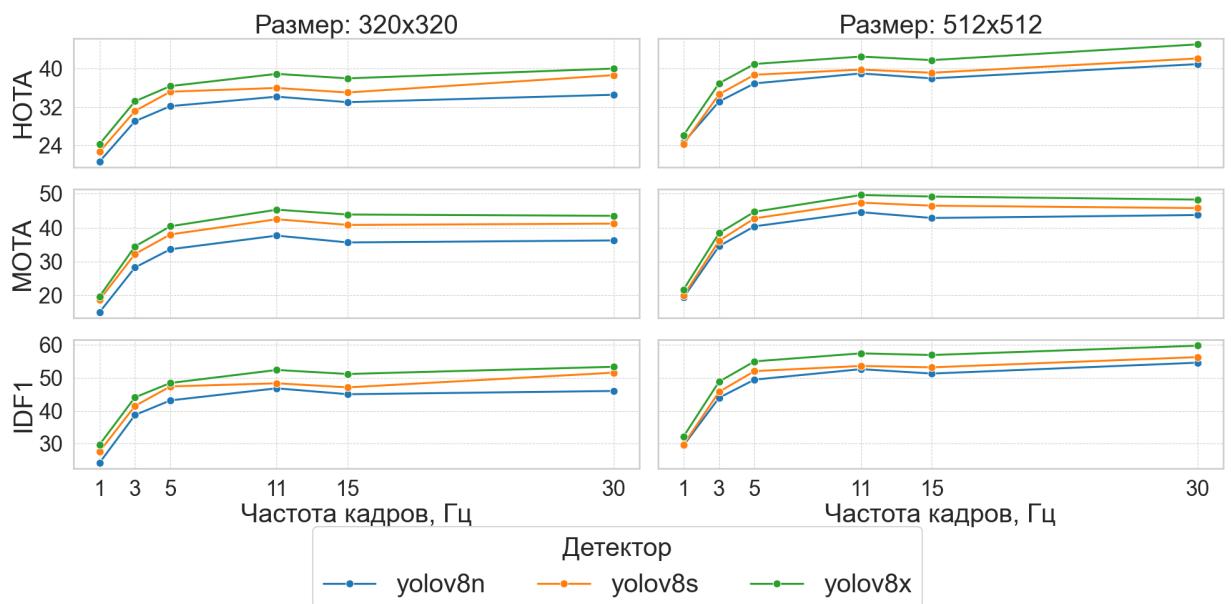


Рисунок 4.15 – График зависимости метрик HOTA, MOTA и IDF1 от частоты кадров видеоизображения для алгоритма Deep OC-SORT

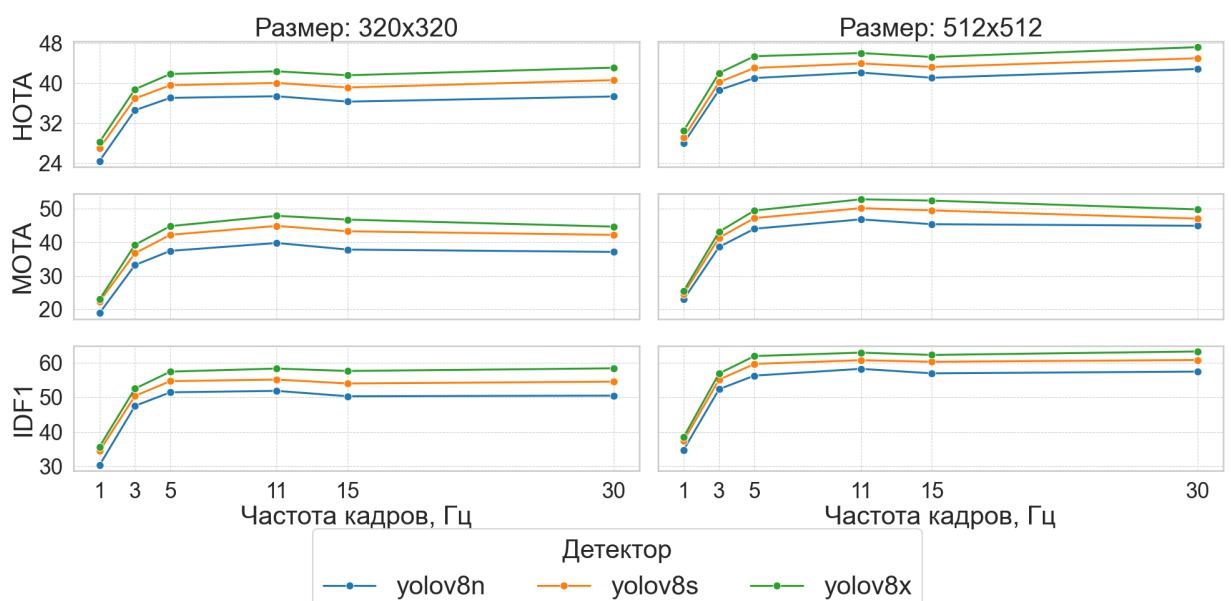


Рисунок 4.16 – График зависимости метрик HOTA, MOTA и IDF1 от частоты кадров видеоизображения для алгоритма StrongSORT

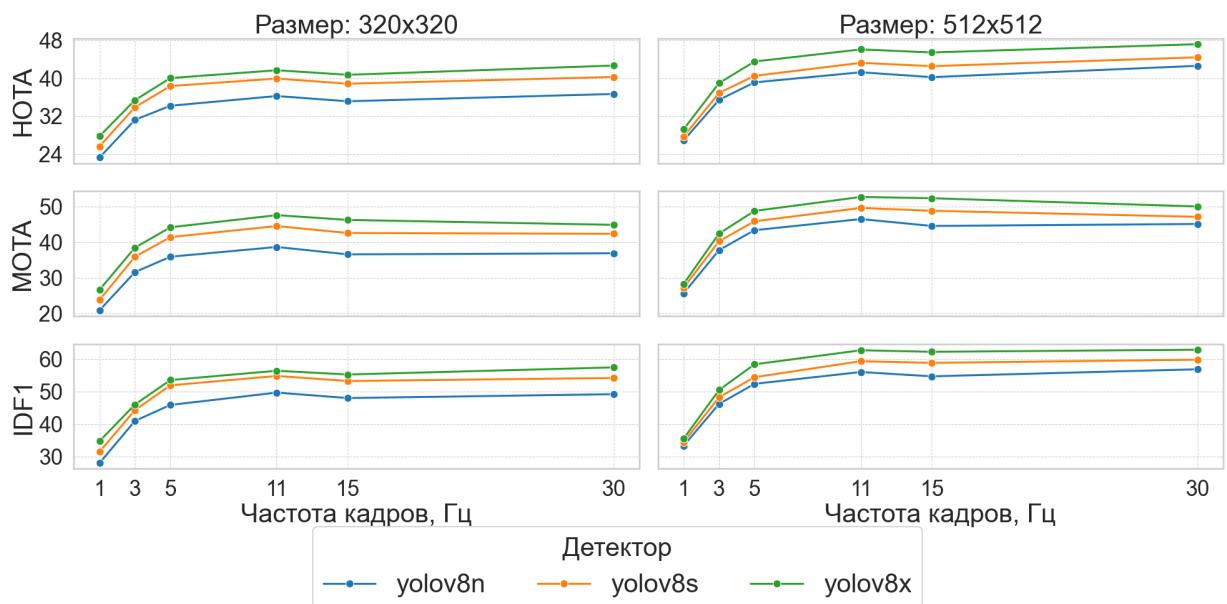


Рисунок 4.17 – График зависимости метрик HOTA, MOTA и IDF1 от частоты кадров видеоизображения для алгоритма BoT-SORT

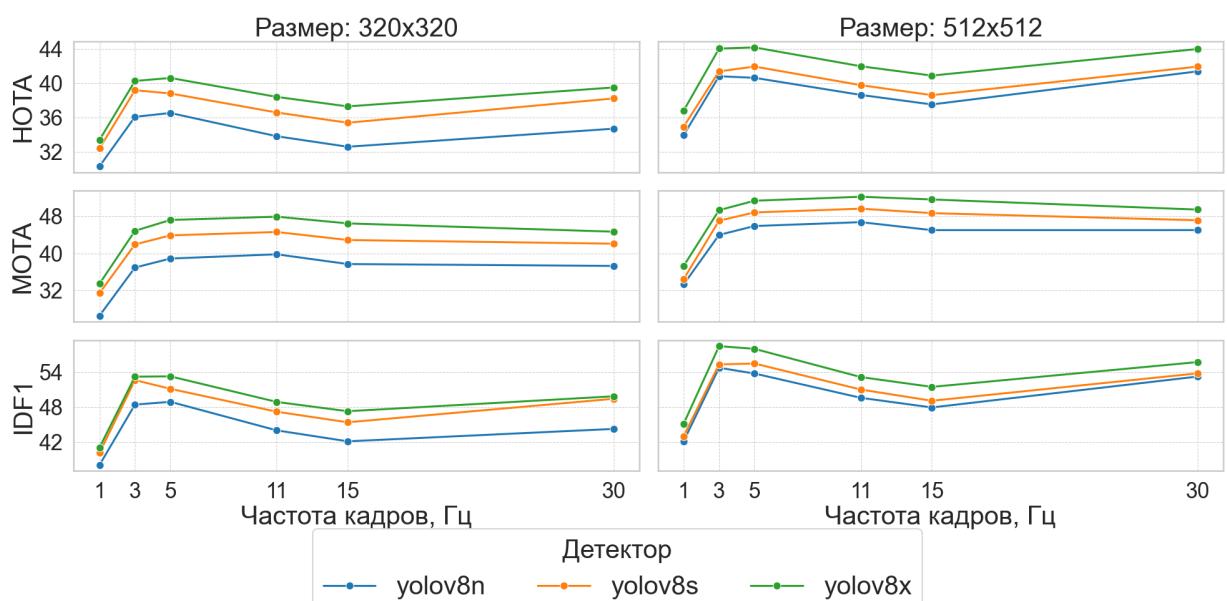


Рисунок 4.18 – График зависимости метрик HOTA, MOTA и IDF1 от частоты кадров видеоизображения для алгоритма ImprAssOC

По полученным результатам четко видна общая закономерность почти для всех методов:

- до 5 кадров в секунду идет значительный линейный рост показателей метрик HOTA, MOTA и IDF1;
- с 5 до 11 происходит небольшое улучшение;
- при частоте выше 11 показатели или выходят на плато, или растут незначительно.

Вероятно, плато является следствием увеличения шумов с ростом частоты кадров видеоизображения, так как перемещения между кадрами становятся все меньше и меньше, а погрешность детектора в несколько пикселей начинает играть все большую роль и перевешивает плюсы от меньших перемещений.

#### **4.4. Исследование зависимости производительности от количества объектов на видеоизображении**

При большом количестве объектов на изображении происходит ощущимая просадка частоты работы при использовании алгоритмов с ReID моделями. Это связано с тем, что их вычисления невозможно перенести на TPU в связи с произвольным размером входа сети. Четвертым экспериментом было решено провести исследование величин этих просадок. Для этого было сделано видео, где на черном фоне один за одним с интервалом в секунду появляются объекты. Результаты запуска различных конфигураций алгоритмов представлены на рисунках 4.19-4.24.

По результатам эксперимента получена таблица 4.5. ByteTrack и OS-SORT – два алгоритма, не использующие ReID-модели – наименее чувствительные к увеличению количества отслеживаемых объектов.

Таблица 4.5 – Корреляция частоты работы с количеством объектов на изображении

Алгоритм	Корреляция
BoT-SORT	-0.63
ByteTrack	-0.03
Deep OC-SORT	-0.65
ImprAssOC	-0.63
OC-SORT	-0.10
StrongSORT	-0.66

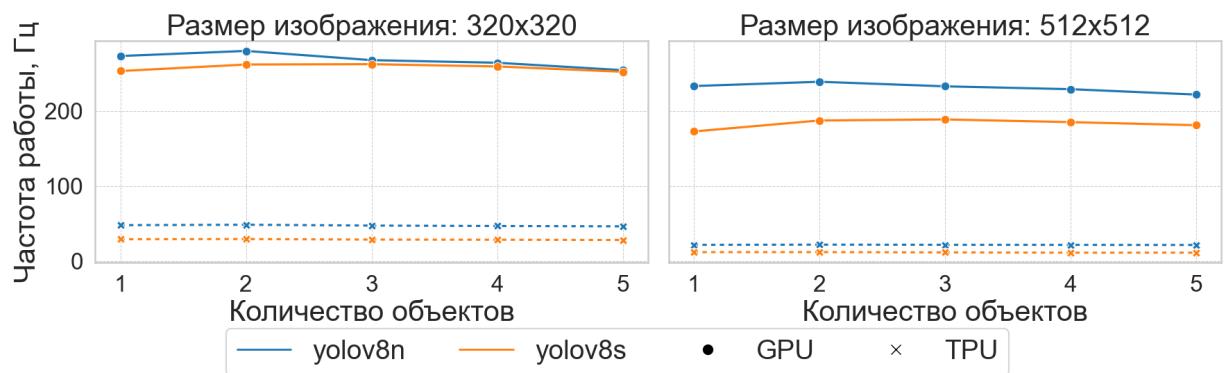


Рисунок 4.19 – График зависимости частоты работы алгоритма ByteTrack от количества объектов на изображении

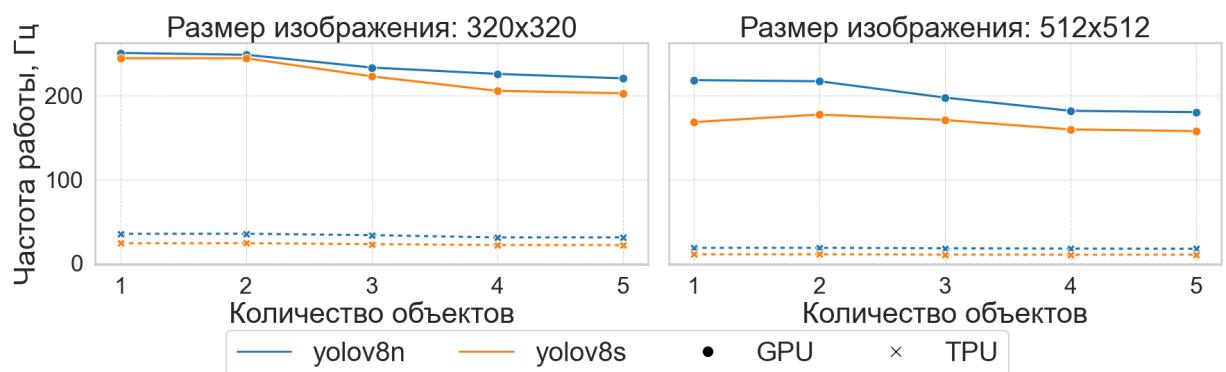


Рисунок 4.20 – График зависимости частоты работы алгоритма OC-SORT от количества объектов на изображении

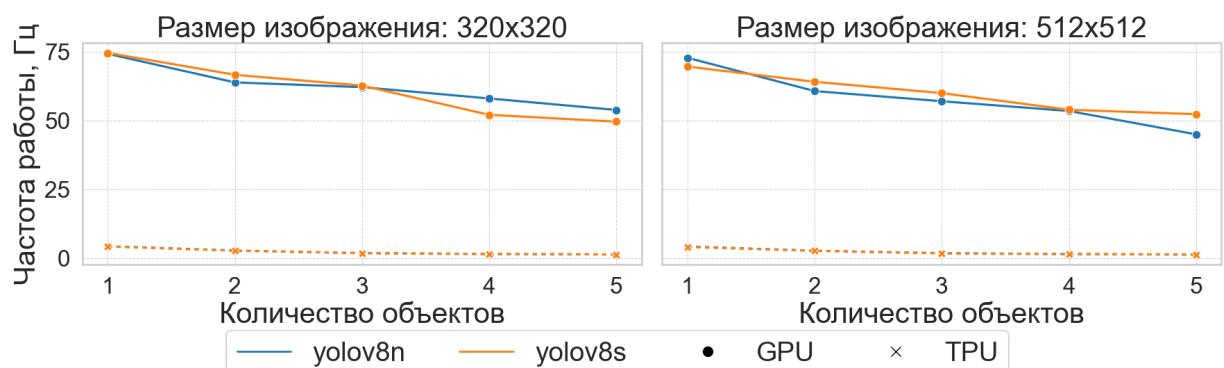


Рисунок 4.21 – График зависимости частоты работы алгоритма Deep OC-SORT от количества объектов на изображении

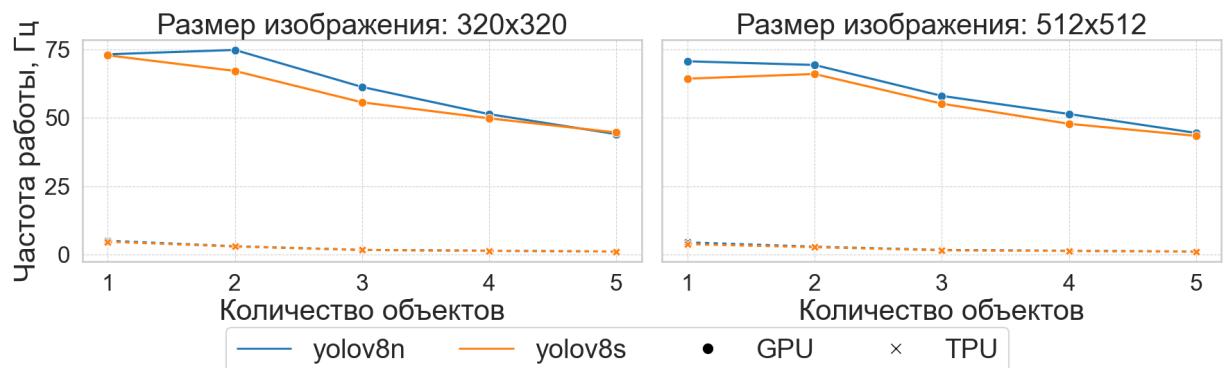


Рисунок 4.22 – График зависимости частоты работы алгоритма StrongSORT от количества объектов на изображении

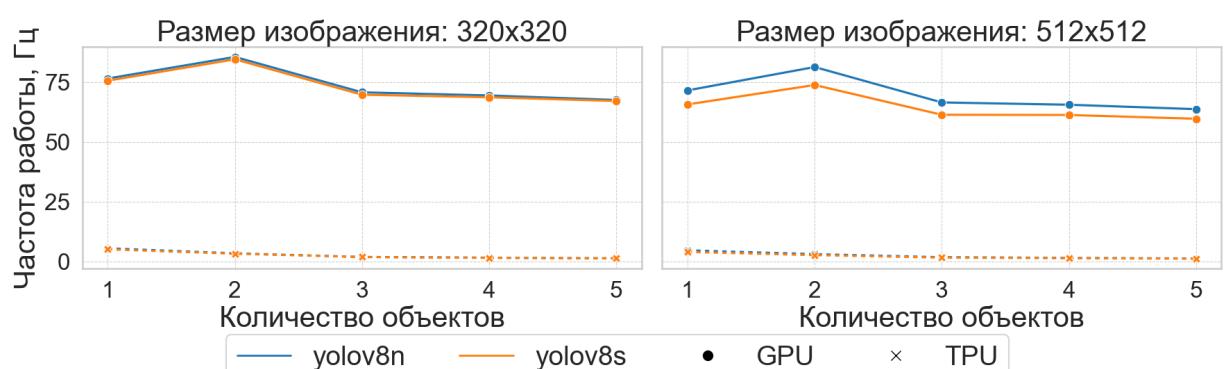


Рисунок 4.23 – График зависимости частоты работы алгоритма BoT-SORT от количества объектов на изображении

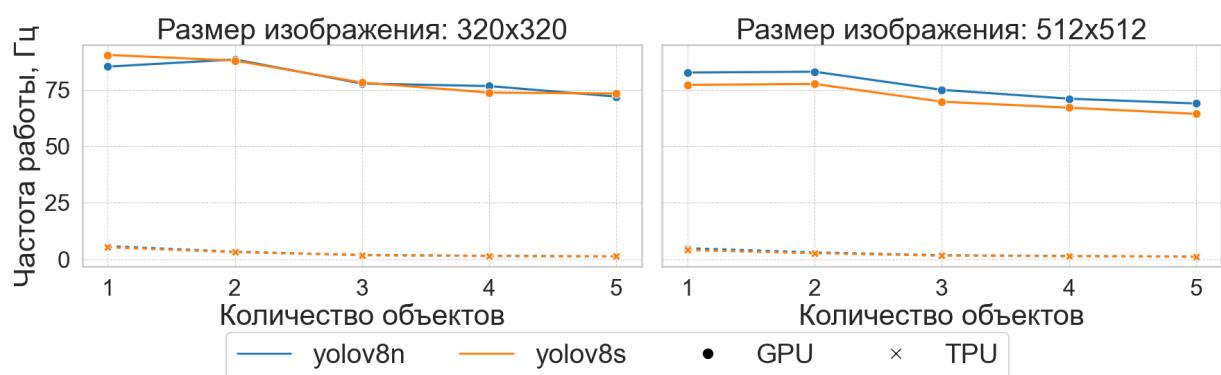


Рисунок 4.24 – График зависимости частоты работы алгоритма ImprAssOC от количества объектов на изображении

## 4.5. Выводы по главе

Результаты всех экспериментов были сведены в таблицы, представленные на рисунках 4.25-4.28 на каждом из которых две таблицы: первая содержит данные о скорости работы различных сочетаний алгоритмов с ReID-моделями на Raspberry Pi 5 с Google Coral TPU; вторая – показатель метрики качества НОТА.

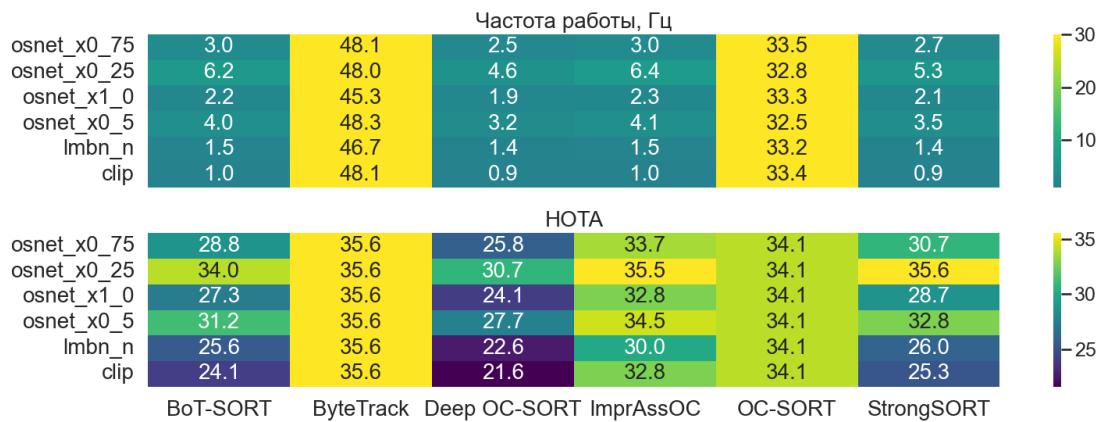


Рисунок 4.25 – Таблица производительности и метрики НОТА для yolov8n с размером изображения 320

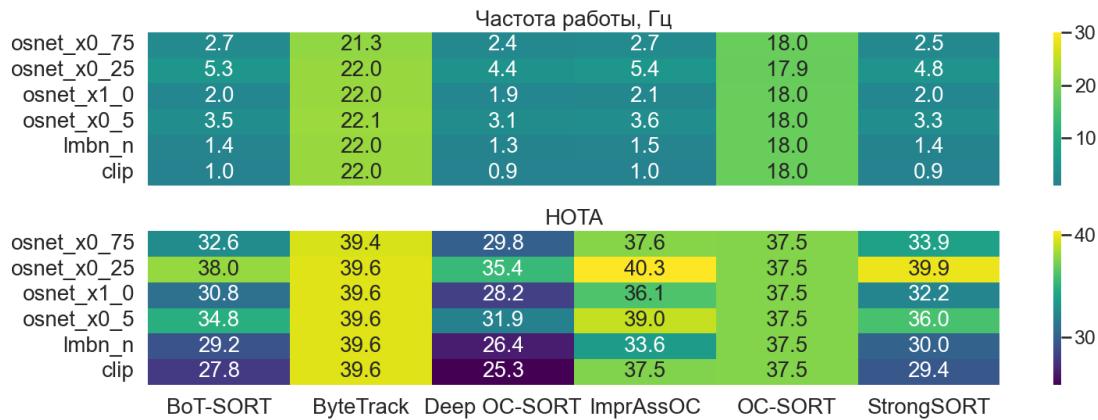


Рисунок 4.26 – Таблица производительности и метрики НОТА для yolov8n с размером изображения 512

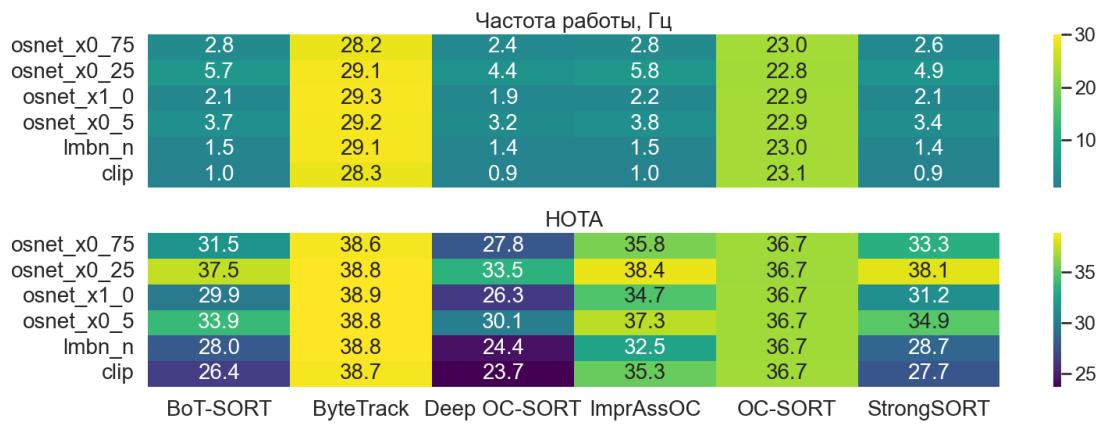


Рисунок 4.27 – Таблица производительности и метрики НОТА для yolov8s с размером изображения 320

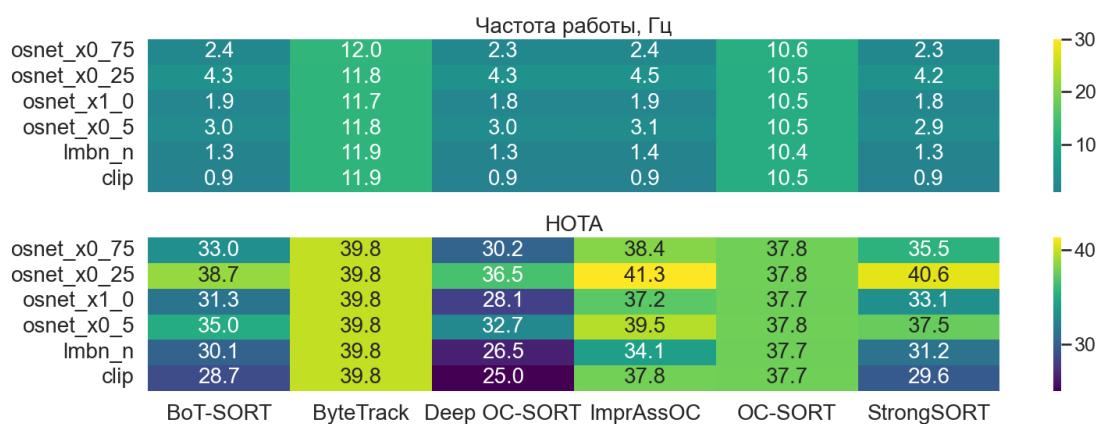


Рисунок 4.28 – Таблица производительности и метрики НОТА для yolov8s с размером изображения 512

## **5 Методические указания на основе результатов экспериментов**

После проведенных экспериментов можно сформировать представленные на рисунках 4.25-4.28 сводные таблицы производительности и качества работы различных алгоритмов на Raspberry Pi 5 с внешним TPU Google Coral.

На основании полученных данных можно сформулировать следующие рекомендации:

- с использованием TPU возможно запустить современные алгоритмы отслеживания объектов на видеоизображениях, но с падением метрик качества до уровня примерно 2020 года;
- за счет более высокой скорости работы yolov8n может успевать работать на размере изображения 512, что дает качество лучше чем yolov8s на 320. При этом yolov8s на 512 начинает работать хуже из-за сильной просадки в скорости работы;
- в общем случае ByteTrack зарекомендовал себя лучше всего. Благодаря нечувствительности к увеличению количества отслеживаемых объектов и общей легковесности, он работает в разы быстрее, за счет чего имеет большое преимущество перед алгоритмами, основывающимися на ReID-моделях;
- OC-SORT обладает всеми теми же плюсами, что и ByteTrack, но работает немного хуже.
- два алгоритма – ImprAssOC и StrongSORT в паре с ReID-моделью OSNet X0.25 – дают наибольшие показатели НОТА, несмотря на низкую скорость работы. Они рекомендуются к использованию в задачах, где количество отслеживаемых объектов не превышает 5, а также скорость передвижения объектов в кадре относительно невысока;
- BoT-SORT и Deep OC-SORT в паре с ReID-моделью OSNet X0.25 дают показатели качества ниже, чем быстрый ByteTrack.

## **ЗАКЛЮЧЕНИЕ**

В ходе работы было проведено исследование 6 современных методов отслеживания объектов на видеоизображениях, а так же выполнен сравнительный анализ с целью выявления наиболее подходящих для низкопроизводительных устройств алгоритмов на основе показателей производительности и метрик. Основными результатами работы являются:

- теоретический обзор современных способов отслеживания объектов на видеоизображениях, набора данных и метрики для сравнения;
- подобраны компоненты для системы апробации;
- проведены эксперименты для каждого из выбранных алгоритмов в различных конфигурациях;
- подготовлены сводные таблицы, демонстрирующие результаты сравнения;
- подготовлены методические рекомендации по выбору алгоритмов в зависимости от задачи.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Exploring the state-of-the-art in multi-object tracking: a comprehensive survey, evaluation, challenges, and future directions / Chenjie Du, Chenwei Lin, Ran Jin et al. // *Multimedia tools and applications*. 2024. Vol. 83. No. 29. P. 73151–73189.
2. Naik B. T., Hashmi M. F., Bokde N. D. A comprehensive review of computer vision in sports: Open issues, future trends and research directions // *Applied Sciences*. 2022. Vol. 12. No. 9. P. 4429.
3. Behavior tracking and analyses of group-housed pigs based on improved bytetrack / Shuqin Tu, Haoxuan Ou, Liang Mao et al. // *Animals*. 2024. Vol. 14. No. 22. P. 3299.
4. Wu P., Li Y., Xue D. Multi-target tracking with multiple unmanned aerial vehicles based on information fusion // *Drones*. 2024. Vol. 8. No. 12. P. 704.
5. Motchallenge: A benchmark for single-camera multiple target tracking / Patrick Dendorfer, Aljosa Osep, Anton Milan et al. // *International Journal of Computer Vision*. 2021. Vol. 129. P. 845–881.
6. Bernardin K., Stiefelhagen R. Evaluating multiple object tracking performance: the clear mot metrics // *EURASIP Journal on Image and Video Processing*. 2008. Vol. 2008. P. 1–10.
7. Performance measures and a data set for multi-target, multi-camera tracking / Ergys Ristani, Francesco Solera, Roger Zou et al. // *European conference on computer vision* / Springer. 2016. P. 17–35.
8. Hota: A higher order metric for evaluating multi-object tracking / Jonathon Luiten, Aljosa Osep, Patrick Dendorfer et al. // *International journal of computer vision*. 2021. Vol. 129. P. 548–578.
9. Bytetrack: Multi-object tracking by associating every detection box / Yifu Zhang, Peize Sun, Yi Jiang et al. // *European conference on computer vision* / Springer. 2022. P. 1–21.
10. Crowdhuman: A benchmark for detecting human in a crowd / Shuai Shao, Zijian Zhao, Boxun Li et al. // *arXiv preprint arXiv:1805.00123*. 2018.
11. A mobile vision system for robust multi-person tracking / Andreas Ess, Bastian Leibe, Konrad Schindler, Luc Van Gool // *2008 IEEE conference on computer vision and pattern recognition* / IEEE. 2008. P. 1–8.

12. Observation-centric sort: Rethinking sort for robust multi-object tracking / Jinkun Cao, Jiangmiao Pang, Xinshuo Weng et al. // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023. P. 9686–9696.
13. Aharon N., Orfaig R., Bobrovsky B.-Z. Bot-sort: Robust associations multi-pedestrian tracking // arXiv preprint arXiv:2206.14651. 2022.
14. Strongsort: Make deepsort great again / Yunhao Du, Zhicheng Zhao, Yang Song et al. // IEEE Transactions on Multimedia. 2023. Vol. 25. P. 8725–8737.
15. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric // 2017 IEEE international conference on image processing (ICIP) / IEEE. 2017. P. 3645–3649.
16. Stadler D., Beyerer J. An improved association pipeline for multi-person tracking // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023. P. 3170–3179.
17. Zhang S., Benenson R., Schiele B. Citypersons: A diverse dataset for pedestrian detection // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. P. 3213–3221.
18. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification / Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, Kris Kitani // 2023 IEEE International Conference on Image Processing (ICIP) / IEEE. 2023. P. 3025–3029.