

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И. Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»

Городецкая Людмила Александровна

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

Образовательная программа 6B06102 – Computer Science

Алматы 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И. Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: «Распознавание твёрдых бытовых отходов в ходе мониторинга
городской агломерации с помощью БПЛА»

Образовательная программа 6B06102 – Computer Science

Выполнила

Городецкая Л.А.

Рецензент

канд. физико-математических наук

ЮН Юничева Н.Р.

« 30 » 05 2025 г.

Научный руководитель

канд. тех. наук., профессор

МУ Мухамедиев Р.И.

« 29 » 05 2025 г.

Алматы 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»

УТВЕРЖДАЮ
Заведующий кафедрой ПИ
канд. тех. наук, ассоц. профессор
Ф.Н. Абдолдина
« 30 » 05 2025 г.



ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся: *Городецкой Людмиле Александровне*

Тема: Распознавание твёрдых бытовых отходов в ходе мониторинга городской агломерации с помощью БПЛА

Утверждена приказом проректора по академической работе:

№ 26-П/Ө

от «29» января 2025 г.

Срок сдачи законченного проекта

« 30 » 05 2025 г.

Исходные данные к дипломному проекту:

- А) Анализ предметной области, обоснование актуальности проекта, определение целей и задач разработки, а также исследование аналогичных проектов;*
- Б) Разработка технического задания;*
- В) Создание набора данных (сбор и разметка изображений);*
- Г) Разработка алгоритмов аугментации данных и обучения нейросетевой модели;*
- Д) Реализация алгоритма для детектирования твёрдых бытовых отходов;*
- Е) Сравнительный анализ точности детектирования при использовании различных наборов аугментации и версий модели YOLO.*

Перечень подлежащих разработке в дипломном проекте вопросов: (с точным указанием обязательных чертежей): *представлены ___ слайда презентации.*

Рекомендуемая основная литература: *из ___ наименований.*

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области	01.12.2024	выполнено
2. Разработка технического задания	01.02.2025	выполнено
3. Создание набора размеченных изображений	20.04.2025	выполнено
4. Разработка подсистемы аугментации	25.04.2025	выполнено
5. Разработка подсистемы обеспечения процессов обучения и верификации качества модели.	01.04.2025	выполнено
6. Выбор и апробирование моделей детектирования	05.04.2025	выполнено
7. Проведение экспериментов с разными версиями модели YOLO и сравнительный анализ точности	20.04.2025	выполнено
8. Написание пояснительной записки к дипломному проекту	25.04.2025	выполнено

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись	Оценка
Программное обеспечение	Еспаев Галымжан Берикович, преподаватель, магистр	28.05.2025		90
Нормоконтролер	Ердалиева Зухра Узакбаевна, преподаватель, магистр	27.05.2025		

Научный руководитель  Мухамедиев Р.И.

Задание принял к исполнению обучающийся  Городецкая Л.А.

Дата « 25 » 05 2025

АҢДАТПА

Қазақстанның қалалық агломерацияларының жедел өсуі қалалар мен қала маңындағы ауа, су және жер үсті ортасының ластануымен байланысты өткір экологиялық проблемаларды тудырады. Қоқыс пен тұрмыстық қалдықтардың стихиялық полигондарының саны артып келеді. Тұрмыстық қатты қалдықтардың (қатты тұрмыстық қалдықтардың) көбеюі, әсіресе урбанизацияланған және өнеркәсіптік аймақтарда күрделі экологиялық проблема болып табылады. Қалдықтарды бақылаудың дәстүрлі әдістері көп уақытты қажет етеді және ауқымы шектеулі. Бұл зерттеу ұшқышсыз ұшу аппараттарынан алынған суреттерді және YOLOv8 терең оқыту моделін пайдалануға негізделген ҚТҚ-ны автоматты түрде анықтау жүйесін ұсынады. Жүйені тестілеу нәтижесінде қатты тұрмыстық қалдықтарды анықтаудың жоғары дәлдігіне қол жеткізілді: дәлдік 84,8%, қалпына келтіру 81,5% және mAP@0.5 (85.1%), IOU 0,5 кезінде. Көрнекі талдау модельдің әртүрлі жағдайларда қалдықтарды минималды қателіктермен тиімді анықтау қабілетін растады. Зерттеу нәтижелері рұқсат етілмеген үйінділерді жедел анықтау және оларды жою шараларын оңтайландыру үшін экологиялық мониторинг жүйелеріне біріктірілуі мүмкін.

АННОТАЦИЯ

Ускоренный рост городских агломераций Казахстана вызывает острые экологические проблемы, связанные с загрязнением воздушной, водной и наземной среды городов и пригородов. Растет число стихийных свалок мусора и бытовых отходов. Увеличение образования твердых бытовых отходов (ТБО) представляет собой серьезную экологическую проблему, особенно в урбанизированных и промышленных регионах. Традиционные методы мониторинга отходов являются трудоемкими и ограничены по масштабам. В этом исследовании предлагается система для автоматического детектирования ТБО, основанная на использовании снимков полученных с борта БПЛА и модели глубокого обучения YOLOv8. В результате тестирования системы была достигнута высокая точность обнаружения ТБО: Precision 84,8%, Recall 81,5% и mAP@0.5 (85.1%), при IoU 0,5. Визуальный анализ подтвердил способность модели эффективно обнаруживать отходы в различных условиях с минимальными ошибками. Результаты исследования могут быть интегрированы в системы экологического мониторинга для оперативного выявления несанкционированных свалок и оптимизации мероприятий по их ликвидации.

ABSTRACT

Accelerated growth of urban agglomerations in Kazakhstan causes acute environmental problems associated with pollution of air, water and land environment of cities and suburbs. The number of unauthorized garbage and household waste dumps is growing. Increased generation of municipal solid waste (MSW) is a serious environmental problem, especially in urbanized and industrial regions. Traditional waste monitoring methods are labor intensive and limited in scope. This study proposes a system for automatic MSW detection based on the use of UAV-derived imagery and YOLOv8 deep learning model. The system was tested and achieved high MSW detection accuracy of Precision 84.8%, Recall 81.5% and mAP@0.5 (85.1%), with an IoU of 0.5. Visual analysis confirmed the model's ability to effectively detect waste under different conditions with minimal errors. The research findings can be integrated into environmental monitoring systems for rapid detection of unauthorized waste dumps and optimization of remediation measures.

СОДЕРЖАНИЕ

Введение	9
1 Анализ существующих технологий мониторинга отходов	10
1.1 Проблема незаконных свалок и её последствия	10
1.2 Традиционные методы обнаружения незаконных свалок	11
1.2.1 Наземный инспекционный контроль	11
1.2.2 Анализ аэрофотоснимков и спутниковых данных	12
1.3 Современные автоматизированные подходы к мониторингу отходов	12
2 Применение YOLO для детекции отходов: теория и обоснование выбора	14
2.1 Принципы работы моделей YOLO	14
2.2 Обоснование выбора моделей YOLO для мониторинга отходов	14
2.3 Сравнение с существующими решениями и обоснование разработки модели	16
3 Особенности разрабатываемой системы	17
3.1 Особенности сбора и подготовки данных	17
3.1.1 Сбор данных	17
3.1.2 Аннотирование данных	18
3.2 Выбор целевого класса объектов и его влияние на эффективность детекции	19
3.3 Критерии оценки точности и скорости детекции	20
4 Разработка и реализация модели детекции отходов	22
4.1 Подготовка аугментированного набора данных для обучения	22
4.2 Подготовка и аннотирование данных с БПЛА	25
4.3 Аугментация данных	27
4.4 Обучение модели	28
4.4.1 Методология эксперимента	28
4.4.2 Сравнительный анализ архитектур	30
4.4.3 Влияние аугментации данных	31
4.5 Подбор гиперпараметров	32
4.6 Адаптация модели для детектирования на Jetson Nano	34
4.7 Визуальная и количественная оценка результатов	35
4.8 Анализ ложных срабатываний модели и их причины	37
4.9 Определение координат объектов на снимках с БПЛА	40
Заключение	44

Список использованной литературы	46
Приложение А. Техническое задание	48
Приложение Б. Листинг программы	50
Приложение В. Описание набора данных	57
Приложение Г. Протокол проведения съёмок искусственно размещённого мусора	58

ВВЕДЕНИЕ

Одной из наиболее значимых экологических проблем современности является рост объёмов твёрдых бытовых отходов (ТБО). Это обусловлено ускоренной урбанизацией и расширением промышленного производства. ТБО включают в себя различные материалы, такие как пластик, металлы, бумага и органические отходы, требующие надлежащей утилизации или переработки. Неэффективная система управления отходами способствует их незаконному размещению. Что в свою очередь приводит к загрязнению почвы и воды, потере биоразнообразия и повышению рисков для здоровья населения. Всё это вызывает необходимость разработки и внедрения современных методов мониторинга отходов.

Традиционно обнаружения незаконных мест скопления ТБО часто выполняются вручную, например, визуальным осмотром мест возможного возникновения свалок. Что характеризуются высокой трудоёмкостью, ограниченным пространственным охватом и низкой оперативностью. В связи с этим **актуальной** задачей становится применение автоматизированных технологий, в частности беспилотных летательных аппаратов (БПЛА) и алгоритмов искусственного интеллекта (ИИ), которые позволяют значительно повысить точность и скорость мониторинга отходов.

Объектом исследования является процесс автоматизированного мониторинга твёрдых бытовых отходов. Предмет исследования – разработка и применение методов детекции ТБО на снимках полученных с борта БПЛА с использованием алгоритмов глубокого обучения.

Целью дипломного проекта является разработка системы автоматического обнаружения ТБО на основе изображений, полученных с БПЛА, и модели YOLO. Для достижения поставленной цели необходимо решить следующие задачи:

1. провести анализ предметной области и существующих методов детекции ТБО;
2. сформировать датасет изображений, содержащий примеры ТБО, выполнить его разметку и подготовку для обучения модели;
3. реализовать алгоритмы аугментации и предобработки данных;
4. провести обучение и тестирование с различными параметрами и аугментациями;
5. оценить точность модели.

В рамках работы будет проведён обзор существующих методов автоматического мониторинга ТБО, а также представлена разработка и тестирование программного обеспечения, реализующего функционал предложенной системы. В заключении будут подведены итоги исследования и обозначены перспективы дальнейшего развития и совершенствования системы автоматического мониторинга твёрдых бытовых отходов.

1 Анализ существующих технологий мониторинга отходов

Проблема незаконных свалок остаётся актуальной из-за их негативного воздействия на окружающую среду, экономику и здоровье населения. Традиционные методы мониторинга, такие как наземные инспекции и анализ спутниковых снимков, обладают рядом ограничений, что затрудняет своевременное выявление очагов загрязнения. В связи с этим актуально внедрение автоматизированных технологий, использующих дроны и алгоритмы машинного обучения.

Настоящий раздел посвящён анализу существующих методов обнаружения незаконных свалок и перспективных подходов к их мониторингу.

1.1 Проблема незаконных свалок и её последствия

Несанкционированные свалки представляют собой скопления отходов, размещённых вне установленных полигонов, что обусловлено недостаточным контролем, несовершенной инфраструктурой утилизации и человеческим фактором. Их наличие влечёт за собой серьёзные экологические, экономические и социальные последствия, требующие системного подхода к решению проблемы.

Экологический ущерб, наносимый незаконными свалками, заключается в загрязнении окружающей среды и усугублении климатических изменений. Согласно данным Всемирного банка, ежегодно в мире образуется около 1,3 млрд тонн твёрдых бытовых отходов, и этот показатель продолжает расти [1]. В странах с низким и средним уровнем дохода управление отходами остаётся недостаточно развитым, что способствует распространению несанкционированных свалок. Такие свалки становятся источником загрязнения почвы и водных ресурсов токсичными соединениями, а также приводят к выбросам метана, образующегося при разложении органических отходов. Метан является мощным парниковым газом, который усиливает глобальное потепление и создаёт риски самовозгорания отходов, что, в свою очередь, может привести к масштабным возгораниям и выбросам вредных веществ в атмосферу [2].

Мегаполисы генерируют значительные объёмы ТБО (до 373 кг/чел. в год), причем их состав включает высокую долю трудноразлагаемых материалов (пластик, стекло). Рост отходов на 14–25% за 5 лет в Санкт-Петербурге, Стокгольме и Хельсинки подтверждает, что урбанизация усугубляет проблему, требуя комплексных решений по переработке и сокращению потребления [3].

В Алматинской агломерации эта проблема приобретает особую остроту: на полигоне захоронения ТБО в селе Айтей Карасайского района, функционирующем с 1990 года, уже накоплено свыше 12 миллионов тонн отходов. При этом ежегодно мусоросортировочный комплекс Green Recycle

принимает около 550 тысяч тонн ТБО, из которых лишь часть (пластик, макулатура, стекло, металлы) направляется на переработку, а неутильные фракции продолжают пополнять объемы захороненных отходов [4].

Проблемы обращения с твёрдыми бытовыми отходами характерны для всей страны: как подчёркивает Бекежанов Д., в Казахстане по-прежнему отсутствует система раздельного сбора ТБО у источника их образования, что затрудняет комплексное управление потоками отходов и снижает эффективность переработки [5].

Экономические последствия незаконных свалок также значительны. Исследование Булековой А. и Габдуллиной С. указывает, что в Западно-Казахстанской области масса накопленных отходов достигает 6,3 млн тонн, а площадь свалок превышает 550 га. Это снижает привлекательность территорий для инвестиций и туризма, а также увеличивает затраты на ликвидацию мусорных очагов. Отсутствие проектной документации и несоблюдение технологий захоронения отходов усугубляют проблему, требуя дополнительных финансовых вложений в инфраструктуру управления ТБО [6].

Социальные последствия незаконных свалок связаны с ухудшением условий жизни населения. Космический мониторинг выявил 503 стихийные свалки только в Уральске и прилегающих районах, что создаёт антисанитарные условия и способствует распространению инфекционных заболеваний. Кроме того, опросы населения показали, что граждане Казахстана готовы поддерживать экологические инициативы по сокращению и переработке отходов, если средства будут направлены на развитие перерабатывающих предприятий [2].

Таким образом, незаконные свалки представляют собой глобальную проблему, которая требует к себе особого внимания. Внедрение современных технологий и методов управления отходами может значительно снизить негативные последствия и способствовать устойчивому развитию урбанизированных территорий.

1.2 Традиционные методы обнаружения незаконных свалок

Помимо автоматизированных систем мониторинга применялись и применяются сейчас различные методы выявления несанкционированных свалок. Например, наземный инспекционный контроль, анализ аэрофотоснимков и спутниковых данных, а также использование дронов с последующей ручной обработкой изображений.

1.2.1 Наземный инспекционный контроль

Метод основан на непосредственном патрулировании территорий сотрудниками экологических служб, а также на поступающих обращениях граждан через горячие линии и специализированные приложения. В процессе выявления нарушений осуществляется фото- и видеосъёмка местности с фиксацией GPS-координат.

В исследовании [7] рассматривается метод прямых наблюдений и измерений, включающий фотографирование объектов и регистрацию их географических координат с использованием мобильных устройств. Несмотря на определённую эффективность, авторы отмечают его высокую трудоёмкость и ограниченный охват территорий, что снижает оперативность и масштабируемость мониторинга.

Статистические данные [8] демонстрируют, что в 2019–2022 гг. в ходе наземных обходов было обнаружено всего 49 несанкционированных свалок, тогда как применение воздушного мониторинга позволило увеличить это число до 715. Это свидетельствует о значительно меньшей эффективности наземного контроля. Также отмечается, что использование БПЛА позволяет выявить все незакрытые густой кроной деревьев свалки ТБО независимо от их расположения, что позволяет существенно дополнять информацию, полученную в ходе наземных маршрутов.

1.2.2 Анализ аэрофотоснимков и спутниковых данных

Данный подход предполагает визуальный анализ изображений с целью выявления мест незаконного размещения отходов, а также использование геоинформационных систем (ГИС) для картографирования свалок.

Согласно исследованию М.В. Семеновой и О.Е. Архиповой («Современные методы мониторинга экологического состояния территории»), мониторинг твёрдых бытовых отходов в Ростовской области осуществляется с применением методов дистанционного зондирования Земли и ГИС. Авторы разработали региональную геоинформационную систему, позволяющую отслеживать полигоны ТБО, вносить новые объекты и анализировать их расположение относительно земель различного целевого назначения. Однако необходимость ручного анализа изображений, зависимость от качества исходных данных и сложность обработки больших массивов информации ограничивают её эффективность [9].

Традиционные методы обнаружения незаконных свалок обладают определённой степенью эффективности, однако их применение сопряжено с высокой трудозатратностью, ограниченным охватом территорий и зависимостью от человеческого фактора. Эти недостатки обуславливают необходимость

разработки и внедрения автоматизированных систем, основанных на современных технологиях.

1.3 Современные автоматизированные подходы к мониторингу отходов

Развитие технологий компьютерного зрения и машинного обучения способствовало появлению более эффективных методов детекции несанкционированных свалок. Применение алгоритмов глубокого обучения позволяет повысить точность и скорость обнаружения отходов на обширных территориях, включая труднодоступные и опасные для человека зоны.

БПЛА широко используются для оценки загрязнения окружающей среды, включая загрязнение воздуха, воды и земельных поверхностей [10].

Помимо задач детектирования с использованием беспилотных летательных аппаратов, в литературе также рассматриваются методы обработки изображений, направленные на анализ данных, полученных с дронов. Например, в исследовании, сравнивающем производительность БПЛА 3DR Iris+ и DJI Phantom 2, были разработаны алгоритмы обработки изображений для оценки устойчивости и точности следования траектории летательных аппаратов по визуальным данным [11]. Снимки, сделанные в ходе зависания и прямолинейного полёта, использовались для расчёта смещений и выявления различий в характеристиках моделей. Это демонстрирует потенциал обработки изображений не только для анализа состояния объектов на земле, но и для улучшения управления и навигации самих БПЛА.

Одним из перспективных направлений является использование беспилотных летательных аппаратов в сочетании с автоматическим анализом полученных изображений. БПЛА позволяют оперативно обследовать большие площади, а алгоритмы машинного обучения автоматически идентифицируют места скопления отходов на аэрофотоснимках. В обзоре [12] рассматриваются современные методы мониторинга мусорных полигонов с применением дронов. Особое внимание уделяется аэрофотосъёмке, позволяющей оперативно собирать детализированные данные о состоянии свалок, оценивать объёмы отходов и контролировать экологическую безопасность. Также говорится, что одним из ключевых преимуществ применения дронов является их способность быстро и с минимальными затратами проводить обследование территорий, снижая риски для персонала.

Эффективность подобных технологий подтверждается пилотным проектом, реализованным в Нур-Султане с октября 2021 года [13]. В рамках которого используются дроны и мобильные роботы для мониторинга незаконных свалок строительных и твёрдых бытовых отходов. Система фиксирует передвижения на территории полигона, распознаёт государственные номера

транспортных средств и передаёт данные в ситуационный центр для дальнейшего анализа и принятия мер.

Внедрение автоматизированных систем мониторинга отходов позволяет повысить оперативность реагирования, снизить трудозатраты и улучшить экологический контроль. Однако их успешная работа требует интеграции с существующими городскими системами видеонаблюдения и оперативного реагирования, а также регулярного обновления алгоритмов обработки данных с учётом специфики местности и сезонных изменений.

2 Применение YOLO для детекции отходов: теория и обоснование выбора

Для автоматизированного мониторинга отходов активно применяются методы компьютерного зрения и глубокого обучения. Среди них особое внимание уделяется моделям семейства YOLO [14, 15, 16]. Данные модели широко используются для задач экологического мониторинга, включая обнаружение твёрдых бытовых отходов на территории городов и природных зон.

В данном разделе рассматриваются принципы работы моделей YOLO, их преимущества перед альтернативными методами детекции, а также обосновывается выбор данной архитектуры для настоящей работы.

2.1 Принципы работы моделей YOLO

Модели семейства YOLO (You Only Look Once) представляют собой одностадийные нейронные сети, предназначенные для детекции объектов в режиме реального времени. В отличие от двухшаговых методов, таких как R-CNN и его модификации, YOLO обрабатывает изображение за один проход, что обеспечивает высокую скорость работы [17].

Основной принцип работы YOLO заключается в разбиении изображения на сетку ячеек, каждая из которых отвечает за предсказание объектов, центры которых находятся внутри неё. Для каждой ячейки сеть прогнозирует координаты ограничивающей рамки (bounding box), вероятность наличия объекта и его класс. Свёрточные нейронные сети (CNN, Convolutional neural network) используются для извлечения признаков, на основе которых полносвязные слои формируют окончательные предсказания. Такой подход обеспечивает баланс между точностью и вычислительной эффективностью.

Благодаря высокой производительности и возможности детекции объектов различных размеров в реальном масштабе времени, YOLO широко применяется в задачах мониторинга окружающей среды, включая автоматизированное обнаружение отходов и контроль загрязнения.

2.2 Обоснование выбора моделей YOLO для мониторинга отходов

Для обоснования выбора конкретной модели был проведён обзор существующих исследований. Так, например, в статье [18] рассматривалась задача детекции мусора на водных поверхностях. В ходе исследования были протестированы такие модели как EfficientDet, SSD, Faster R-CNN, RetinaNet и

Yolov8l. Последняя продемонстрировала наилучшие результаты, при сравнении следующих метрик: точность (Precision) и коэффициент Жаккара (Jaccard Index).

Высокая точность обнаружения была достигнута в исследовании [19]. В этой статье представлена модель классификации и детекции отдельных объектов ТБО (пластиковых пакетов, бутылок и т.д.). YOLOv5s в данной работе превзошла Faster R-CNN по точности, а также продемонстрировала трёхкратное превосходство по скорости детекции. В целом, это исследование демонстрирует эффективность применения БПЛА в сочетании с глубоким обучением для автоматического мониторинга пластиковых отходов. Разработанная система достигла точности свыше 85% при обнаружении плавающего мусора и обеспечила скорость свыше 40 FPS, что делает её применимой для задач реального времени.

В статье [20] рассматривается проблема очередей и возможность её решения с помощью методов машинного обучения. Основное внимание уделяется задаче подсчёта людей с использованием моделей компьютерного зрения. Для этого авторы проводят сбор и аннотирование специализированного набора данных, содержащего изображения с камер видеонаблюдения. В качестве инструментов детекции анализируются две популярные архитектуры: одноэтапная YOLOv5 и двухэтапная Faster R-CNN. В ходе исследования проведено обучение и валидация моделей, а также их сравнение по ключевым метрикам — точности (precision), полноте (recall), средней точности (AP) и скорости обработки (FPS). Эксперименты показали, что YOLOv5 значительно превосходит Faster R-CNN по скорости (57 FPS против 7 FPS), что делает его более подходящим для обнаружения объектов в реальном времени. При этом показатели точности у YOLOv5 оказались ниже всего на 1-2%.

Таким образом, выбор моделей YOLO для задач мониторинга отходов является обоснованным решением. Так как эти модели сочетают в себе высокую точность детектирования и способность обрабатывать кадры с высокой частотой. Также они успешно применяются в реальных проектах, направленных на улучшение управления отходами и защиту окружающей среды. Дополнительным плюсом является эффективность на устройствах с ограниченными вычислительными ресурсами. Это позволяет интегрировать решение на платформу Jetson Nano и использовать его для детекции мусора с беспилотного летательного аппарата.

2.3 Сравнение с существующими решениями и обоснование разработки собственной модели

В исследовании [21] представлена система автоматического детектирования отходов с применением моделей семейства YOLO, развёрнутая на беспилотном летательном аппарате с использованием платформы NVIDIA Jetson Nano. Решение обеспечивало обработку изображений в реальном времени,

анализ поступающих данных и передачу информации о выявленных объектах на удалённые станции в виде тепловых карт. В ходе экспериментов были протестированы различные версии YOLO, оценённые по точности и скорости работы. Результаты показали высокую эффективность метода по сравнению с традиционными подходами.

Тем не менее при практическом тестировании модели возникли сложности, обусловленные условиями съёмки. Испытания проводились на трёх локациях:

1. **кампус NTOU** – демонстрировалось высокое качество детекции с точностью до 98%;

2. **рыболовный порт Badouzi** – наблюдалось значительное снижение точности, вызванное изменением текстуры отходов под воздействием воды и эрозии;

3. **пляж Wanghai Xiang** – выявлены трудности в различении мусора и природных объектов (например, камней и песка), что приводило к увеличению количества ложных срабатываний.

Данные результаты подчёркивают необходимость адаптации модели к специфическим условиям съёмки и видам обнаруживаемых объектов, что обосновывает целесообразность разработки собственной модели.

Другая практическая реализация предложена в работе [22]. В ней разработана система учёта мусора с использованием БПЛА и модели YOLOv5. Методология включает применение веб-приложений для визуализации выявленных отходов на цифровых картах, что способствует оптимизации управления твёрдыми отходами. Однако данное решение имеет ограничения, связанные с отсутствием анализа влияния различных условий съёмки (освещения, фона, погодных факторов), что может снижать точность детекции в сложных сценариях.

Разрабатываемая в данной работе система учитывает преимущества существующих решений, но при этом дополняется за счёт использования уникального набора данных, собранного на территории Алматинской городской агломерации. Это позволит модели адаптироваться к специфическим условиям региона, включая сезонные изменения (например, наличие снежного покрова зимой) и различные погодные условия. Важным отличием предложенного подхода является фокус не на детекции отдельных видов отходов (бутылки, пластиковые пакеты и пр.), а на выявлении свалок как единого объекта. Данный аспект требует особого внимания при разработке модели и её обучении, что также обосновывает необходимость создания специализированного решения.

3 Особенности разрабатываемой системы

Разработка системы автоматического распознавания твёрдых бытовых отходов с применением методов компьютерного зрения требует комплексного подхода к созданию и оптимизации модели. Основным компонентом такой системы является глубокая нейросеть, способная эффективно детектировать объекты на изображениях, полученных с БПЛА. Однако точность и надёжность работы модели зависят не только от её архитектуры, но и от качества входных данных, методов их разметки и параметров обучения.

В данном разделе рассматриваются ключевые аспекты разработки системы, включая процесс сбора данных, аннотирования изображений. Также анализируются особенности выбора целевого класса «свалка» и факторы, влияющие на эффективность детекции. Кроме того, описаны критерии оценки модели, которые позволяют объективно измерять её точность и производительность в реальном времени.

Представленный подход направлен на создание высокоточной системы, способной функционировать в разнообразных условиях и эффективно выявлять несанкционированные свалки.

3.1 Особенности сбора и подготовки данных

Качество данных играет ключевую роль в эффективности моделей компьютерного зрения. Для успешного обучения модели YOLO для задачи детекции ТБО необходимо обеспечить разнообразие и точность разметки изображений. В данном разделе рассматриваются основные этапы подготовки данных, влияющие на точность и обобщающую способность модели.

3.1.1 Сбор данных

На первом этапе осуществляется сбор данных для обучения, валидации и тестирования модели. Качество и разнообразие данных критически важны для обеспечения высокой производительности модели компьютерного зрения, поскольку они определяют её способность к обобщению и корректному распознаванию объектов в различных условиях.

Для этого необходимо проводить съёмку в несколько этапов, охватывая сезонные изменения (наличие снега, травы, листвы и т. д.), а также различные локации. В рамках исследования предлагается использовать изображения санкционированных свалок строительного мусора в урбанизированных районах, поскольку их сбор наиболее доступен и экономически оправдан. Дополнительно

в набор данных включены изображения несанкционированных свалок в природных зонах Алматинской области. Для увеличения объёма и вариативности данных также неоднократно были проведены съемки бытового мусора, искусственно размещенного в различных местах.

Собранные данные обладают высокой долговременной ценностью и могут использоваться повторно в последующих исследованиях.

3.1.2. Аннотирование данных

Вторым этапом подготовки данных является разметка изображений, полученных с дронов. Этот процесс включает ручное аннотирование с использованием специализированных инструментов. Аннотации играют ключевую роль в обучении модели, так как определяют, какие объекты она должна распознавать. Чем больше размеченных изображений содержит набор данных, тем точнее модель будет идентифицировать объекты заданного класса.

Разметка изображений осуществляется с помощью различных инструментов, позволяющих выделять границы целевых объектов. Чаще всего используются ограничивающие рамки, которые очерчивают объекты в пределах изображения. Среди наиболее удобных сервисов для аннотирования данных выделяется **Roboflow**, предоставляющий интуитивно понятный интерфейс и широкий спектр обучающих материалов.

В одной из статей [23] отмечается, что наряду с получением изображений, максимально приближенных к реальным условиям, особое внимание следует уделять точности разметки. Так как качество аннотаций напрямую влияет на точность модели, важно соблюдать определённые принципы при разметке.

1. Аннотирование всех целевых объектов на изображении.

Каждое появление объекта в кадре должно быть отмечено. Пропуск объектов может привести к формированию ложноотрицательных примеров, что снизит точность модели.

2. Выделение объекта полностью.

Ограничивающая рамка должна включать весь объект целиком. Частичное аннотирование может исказить представление модели о границах объекта.

3. Разметка частично закрытых объектов.

Если объект частично скрыт другими элементами сцены (окклюзия), его всё равно следует аннотировать. Оптимальной стратегией является разметка объекта так, как если бы он был полностью видимым.

4. Создание точных ограничивающих рамок.

Рамки должны плотно прилегать к объекту, но не обрезать его части. Чрезмерно широкие рамки могут включать посторонние пиксели, что ухудшает качество обучения.

Применение строгих методологических подходов при аннотировании позволяет создать высококачественный набор данных, способствующий повышению точности и надежности модели компьютерного зрения.

3.2 Выбор целевого класса объектов и его влияние на эффективность детекции

В рамках данного исследования целевым классом объектов является «свалка» (*dump*). Он включает в себя санкционированные и несанкционированные скопления твёрдых отходов. Особенности этого класса оказывают значительное влияние на эффективность детекции.

Ниже приведены основные факторы, влияющие на точность модели.

1. Разнообразие состава и структуры

Свалки содержат отходы различных типов (строительные материалы, пластик, бумага, органические остатки и пр.), что приводит к высокой вариативности визуальных признаков.

2. Размер и форма

Масштабы свалок могут варьироваться от локальных загрязнений до крупных полигонов, что требует от модели способности обнаруживать объекты разного размера.

3. Фоновая среда и контрастность

Свалки располагаются как в урбанизированных, так и в природных зонах, что влияет на различимость объекта относительно окружающего фона. В условиях низкого контраста возможны ошибки детекции.

4. Вариативность условий съёмки

Освещённость, время суток, наличие теней и сезонные изменения (например, снег или растительность) могут существенно изменять визуальные характеристики свалки, затрудняя её распознавание.

Таким образом, для повышения точности модели при обнаружении свалок необходимо:

- **Формировать разнообразный набор данных.** Датасет должен включать изображения свалок разного состава и размера. Снимки необходимо делать в различных локациях при разных погодных условиях.

- **Применять аугментацию данных,** чтобы имитировать сложные сценарии (изменения освещения, частичная окклюзия объектов и пр.), повышая устойчивость модели к внешним факторам.

Такой подход позволит улучшить обобщающую способность модели и повысить эффективность детекции свалок в реальных условиях.

3.3 Критерии оценки точности и скорости детекции

Оценка качества обученной модели является важнейшим этапом наравне со сбором датасета и обучением. Этот этап позволяет выбрать оптимальный вариант при экспериментах с различными гиперпараметрами, методами аугментации и архитектурными изменениями. Для определения эффективности детекции объектов в YOLO используются метрики, характеризующие как точность модели, так и её производительность в режиме реального времени [24].

Метрики точности

1. Точность (Precision):

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

где TP – количество корректно предсказанных объектов, FP – ложные срабатывания.

2. Полнота (Recall):

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

где FN – количество пропущенных объектов.

3. **Средняя точность** (mAP, Mean Average Precision) – ключевой показатель, оценивающий качество детекции на основе Precision-Recall кривой. Он рассчитывается как среднее значение Average Precision (AP) по всем классам объектов.

В YOLO наиболее часто применяются две модификации этой метрики:

- mAP0.5 (mAP50) – рассчитывается при пороге IoU = 0.5;
- mAP0.5:0.95 – усреднённое значение AP при изменении порога IoU в диапазоне от 0.5 до 0.95 с шагом 0.05, что является более строгим критерием.

4. **Пересечение над объединением** (IoU, Intersection over Union) – показатель совпадения предсказанных ограничивающих рамок с эталонными разметками (ground truth):

$$IoU = \frac{\text{площадь пересечения}}{\text{площадь объединения}}.$$

Для успешного обнаружения объекта пороговое значение $IoU > 0.5$ считается удовлетворительным, однако в высокоточных задачах могут применяться более строгие критерии.

Метрики производительности

1. **Кадры в секунду** (FPS, Frames Per Second) – число изображений, обрабатываемых моделью за одну секунду.

Высокопроизводительные версии YOLO (например, YOLOv8) могут достигать **>100 FPS** на современных графических ускорителях.

2. Время инференса (Inference Time) – время, необходимое для обработки одного изображения.

Этот параметр критически важен для задач реального времени, таких как видеоаналитика или управление автономными системами.

Стоит также отметить, что помимо количественных характеристик, оценка качества модели должна включать визуальный анализ детекций. Этот метод позволяет выявить типичные ошибки модели, включая ложные срабатывания и пропуски объектов в различных условиях. Анализ таких ситуаций способствует дальнейшему улучшению модели и её адаптации к специфическим сценариям применения.

Функция потерь модели

Обучение YOLOv8 оптимизируется с помощью суммарной функции потерь, включающей три компонента:

`box_loss` — отвечает за точность предсказания координат ограничивающих рамок (bounding boxes);

`cls_loss` — определяет ошибки в предсказании классов объектов;

`dfl_loss` (Distribution Focal Loss) — улучшает прогнозирование границ объектов за счёт более точного распределения вероятностей

Уменьшение этих значений при обучении также указывает на то, что он движется успешно.

4 Разработка и реализация модели детекции отходов

4.1 Подготовка аугментированного набора данных для обучения

Как было показано в исследовательско-технологическом разделе, одним из ключевых факторов успешного обучения нейросетевой модели для детекции ТБО является наличие разнообразного и репрезентативного обучающего набора данных. Разнообразие в визуальных характеристиках объектов, погодных условиях, фонах и масштабах позволяет значительно повысить способность модели к обобщению и точному распознаванию объектов в реальных условиях эксплуатации.

В рамках данного исследования задача формирования такого набора данных была решена посредством организации и проведения нескольких этапов съёмки. Подготовка данных включала в себя как съёмку реально существующих свалок, так и создание условий, моделирующих загрязнение окружающей среды.

Съёмка санкционированной свалки строительного мусора

Одним из первых источников данных стали изображения санкционированной свалки строительного мусора, расположенной в непосредственной близости от университета (улица Каныша Сатпаева, 22). Съёмка проводилась в разные сезоны, что позволило зафиксировать изменения внешнего вида объекта в зависимости от погодных условий. Примеры полученных изображений представлены на рисунках 4.1.1 и 4.1.2.



Рисунок 4.1.1 – свалка строительного мусора зимой



Рисунок 4.1.2 – свалка строительного мусора летом

Искусственное размещение и съёмка бытового мусора

Для моделирования ситуаций, близких к реальным несанкционированным свалкам, был проведён этап искусственного размещения бытового мусора с последующей фото- и видеофиксацией. Для этого нами был собран типичный мусор из повседневной жизни: пластиковые бутылки, алюминиевые банки, картон, бумажные отходы, упаковки от пищевых продуктов и прочее.

Размещение осуществлялось с регулярностью один раз в неделю или раз в две недели в двух различных локациях с характерным ландшафтом.

- Первая локация была окружена высокими деревьями, чьи ветви частично закрывали размещённый мусор. Это позволяло имитировать сложные сцены с перекрытиями, а также учить модель отличать мусор от растительных элементов. Ранние версии модели демонстрировали ошибочные срабатывания на фоне деревьев, что и стало основанием для включения таких изображений в датасет.

- Вторая локация представляла собой открытую площадку с несколькими невысокими елями и кустарником. Такая сцена позволяет модели адаптироваться к фонам с более сложной структурой и учиться избегать ложных положительных срабатываний.

Съёмка проводилась в разные погодные и сезонные условия, включая:

- полное снежное покрытие,
- частичное таяние снега,
- прошлогоднюю листву,
- весенний травяной покров.

Таким образом, был охвачен широкий спектр визуальных вариаций, встречающихся в реальной среде. Пример изображения искусственно размещённого мусора приведён на рисунке 4.1.3. Полный протокол съёмок и дополнительные изображения представлены в приложении 1.



Рисунок 4.1.3 – искусственно размещённый мусор рядом с университетом в апреле

Съёмка несанкционированных свалок в природных зонах

Наиболее трудоёмким и важным этапом стало получение изображений реальных несанкционированных свалок, расположенных в природных зонах Алматинской области. Эти съёмки были технически сложными: свалки было трудно обнаружить, они находятся на значительном удалении от городской черты.

Однако включение подобных изображений в обучающий датасет является критически важным, поскольку такие сцены радикально отличаются от изображений городских свалок или искусственно смоделированных участков. Мусор в природной среде часто частично погружается в почву, покрывается растительностью, обесцвечивается под воздействием солнечного света и осадков. Все эти факторы делают его визуально менее контрастным и труднее поддающимся распознаванию (см. рисунок 4.1.4).



Рисунок 4.1.4 – свалка в горах Алматинской области весной.

Таким образом, добавление подобных изображений позволяет повысить устойчивость модели и её применимость к реальным задачам мониторинга.

4.2 Подготовка и аннотирование данных с БПЛА

Для подготовки входных данных, необходимых для обучения нейронной сети, каждая видеозапись, полученная с борта БПЛА, была преобразована в набор кадров с разрешением 1920×1080 пикселей.

В состав выборки включались как кадры, содержащие твёрдые бытовые отходы (ТБО), так и изображения, на которых мусор отсутствовал. Это было сделано с целью улучшения способности модели различать целевые объекты и фон, а также для снижения количества ложных срабатываний при последующей детекции.

В результате этого этапа был сформирован датасет, состоящий из 2000 изображений (1500 изображения, содержащие ТБО, и 500 фонов). Далее весь объём данных был разделён на три подвыборки согласно стандартной пропорции:

- **Обучающая выборка (70%)** - использовалась для оптимизации параметров модели в процессе обучения;
- **Валидационная выборка (20%)** - применялась для оценки качества модели на неизменяемых данных во время тренировки;
- **Тестовая выборка (10%)** - использовалась для окончательной проверки и визуальной оценки способностей модели.

Разметка изображений для обучающей и валидационной подвыборок выполнялась вручную с использованием специализированного веб-сервиса Roboflow (<https://app.roboflow.com>). На каждом изображении все обнаруженные свалки ТБО выделялись ограничивающими прямоугольниками (bounding boxes), что позволяло задать точные координаты объектов для последующего использования в процессе обучения модели.

Пример пользовательского интерфейса разметки представлен на рисунке 4.2.1.

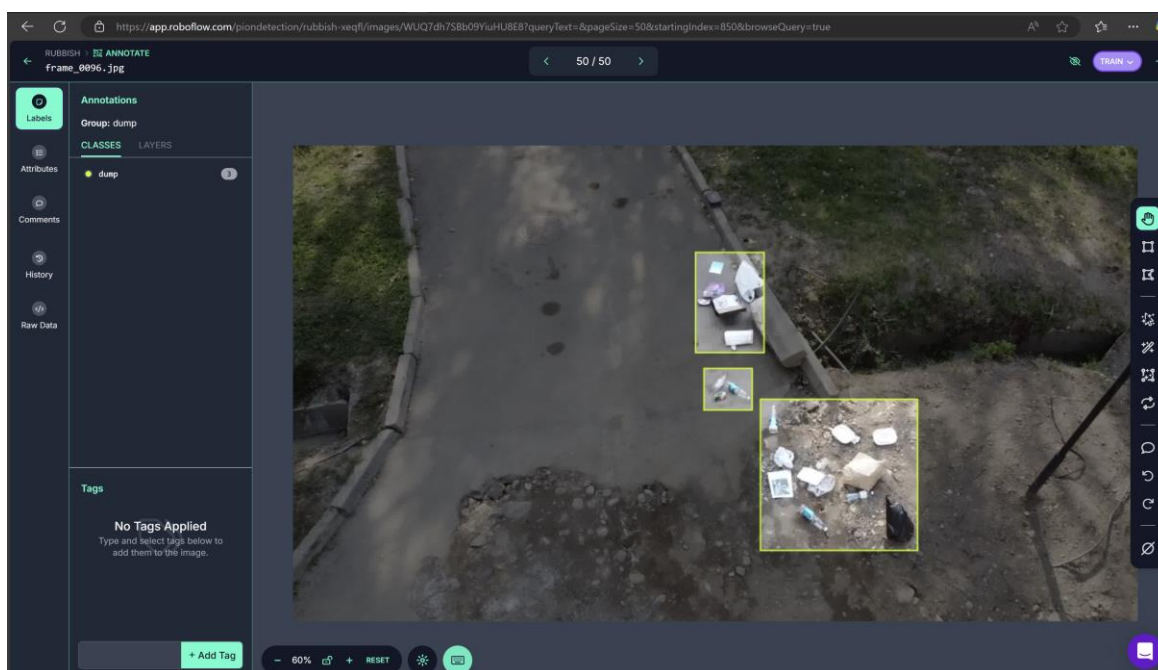


Рисунок 4.2.1 – Разметка данных с помощью инструмента Roboflow

После завершения разметки Roboflow автоматически генерирует файлы аннотаций, которые могут быть экспортированы в различных форматах, включая формат, совместимый с архитектурой YOLOv8. Структура экспортируемого проекта при этом имеет следующий вид: три папки — train, valid и test, а также конфигурационный файл data.yaml. В последнем указываются пути к соответствующим папкам с изображениями и аннотациями, а также словарь классов, в котором перечислены все категории объектов и их индексы. В рамках данного исследования используется только один класс — 0: "dump" (свалка).

Каждая из трёх основных папок (train, valid, test) содержит по две вложенные директории: images и labels. Папка images включает исходные изображения в формате .jpg, ранее загруженные и размеченные в Roboflow. В папке labels находятся текстовые файлы с аннотациями в формате .txt.

Каждый файл аннотаций соответствует одному изображению и содержит по одной строке на каждый объект (свалку), обнаруженный на изображении.

Каждая строка состоит из пяти числовых значений, следующих в определённом порядке: первое значение — индекс класса (в нашем случае всегда

0), а следующие четыре значения представляют собой нормализованные координаты ограничивающего прямоугольника (`x_center`, `y_center`, `width`, `height`). Все координаты нормализованы, то есть выражены в долях от ширины и высоты изображения и находятся в диапазоне от 0 до 1. Такой формат облегчает обработку данных моделью и делает её независимой от абсолютного разрешения изображений.

4.3 Аугментация данных

Для повышения обобщающей способности модели и увеличения объёма обучающего датасета была проведена процедура аугментации изображений. Аугментация позволяет сгенерировать дополнительные вариации исходных изображений и соответствующих разметок, что особенно важно при наличии ограниченного количества оригинальных данных.

В данной работе для реализации аугментации была использована библиотека `imgaug`, обеспечивающая гибкое и наглядное применение разнообразных трансформаций как к изображениям, так и к соответствующим ограничивающим рамкам. Основные этапы этого процесса представлены ниже.

Конфигурация аугментатора

Аугментатор был реализован с помощью функции `create_augmenter()`, в которой определена последовательность возможных преобразований:

- масштабирование (от 90% до 110% от исходного размера),
- горизонтальное и вертикальное отражение (`FlipLR`, `FlipUD`),
- поворот изображения в диапазоне от -15° до $+15^\circ$,
- размытие по Гауссу (`GaussianBlur`),
- изменение яркости посредством умножения значений пикселей (`Multiply`).

Трансформации применяются в случайном порядке, что увеличивает разнообразие аугментированных данных.

Применение аугментации к изображениям и разметкам

Аугментация применяется не только к изображениям, но и к соответствующим аннотациям в формате YOLO. Сначала координаты ограничивающих рамок преобразуются в абсолютные значения относительно размеров изображения, после чего они передаются в `BoundingBoxesOnImage`, объект библиотеки `imgaug`, который обеспечивает корректное применение всех преобразований. После выполнения аугментации происходит отбрасывание выходящих за границы изображения рамок, а оставшиеся обрезаются по границам. Затем координаты преобразуются обратно в относительный YOLO-формат и сохраняются.

Многократное применение и сохранение результатов

Для генерации заданного количества аугментированных образцов используется функция `run_augmentation`, в рамках которой:

- случайным образом отбираются изображения и соответствующие метки из обучающей выборки;
- для каждого образца инициализируется отдельный экземпляр аугментатора с уникальным seed;
- выполняется аугментация и визуализация результата;
- изображения и аннотации сохраняются в соответствующие директории (train/images и train/labels).

Данный подход позволил значительно расширить исходную обучающую выборку без потери качества аннотаций, что в дальнейшем может обеспечить модели дополнительную устойчивость к изменению масштаба, ориентации и освещённости объектов на изображениях и повысить её точность.

4.4 Обучение модели

4.4.1 Методология эксперимента

В рамках исследования проведен сравнительный анализ эффективности трёх различных архитектур семейства YOLOv8 (s, m, x) на специализированном наборе данных. Экспериментальная выборка включает:

- 1470 изображений для обучения (включая 439 фоновых),
- 425 изображений для валидации (включая 123 фоновых).

Обучение модели YOLOv8 происходит с использованием метода переноса обучения (transfer learning). Этот подход был выбран как оптимальный из-за ограниченного объёма доступных данных, полученных с БПЛА.

Использование предварительно обученных весов позволяет модели задействовать знания, полученные в процессе предварительного обучения на большом наборе данных COCO (Common Objects in Context) [25], тем самым ускоряя процесс адаптации к новой задаче и снижая вероятность переобучения.

Для части экспериментов осуществлено искусственное расширение обучающей выборки на 1000 синтетически модифицированных изображений с применением методов аугментации данных.

Параметры обучения:

- продолжительность: 60 эпох (эмпирически установленный оптимум);
- аппаратная платформа: NVIDIA GeForce RTX 3060.

Как демонстрирует Рисунок 4.4.1, графики функций потерь (loss) и ключевых метрик (precision, recall, mAP) достигают плато к 60-й эпохе, что свидетельствует о сходимости процесса обучения. Дальнейшее увеличение количества эпох нецелесообразно ввиду: отсутствия значимого улучшения качества, риска переобучения и неоправданного роста вычислительных затрат.

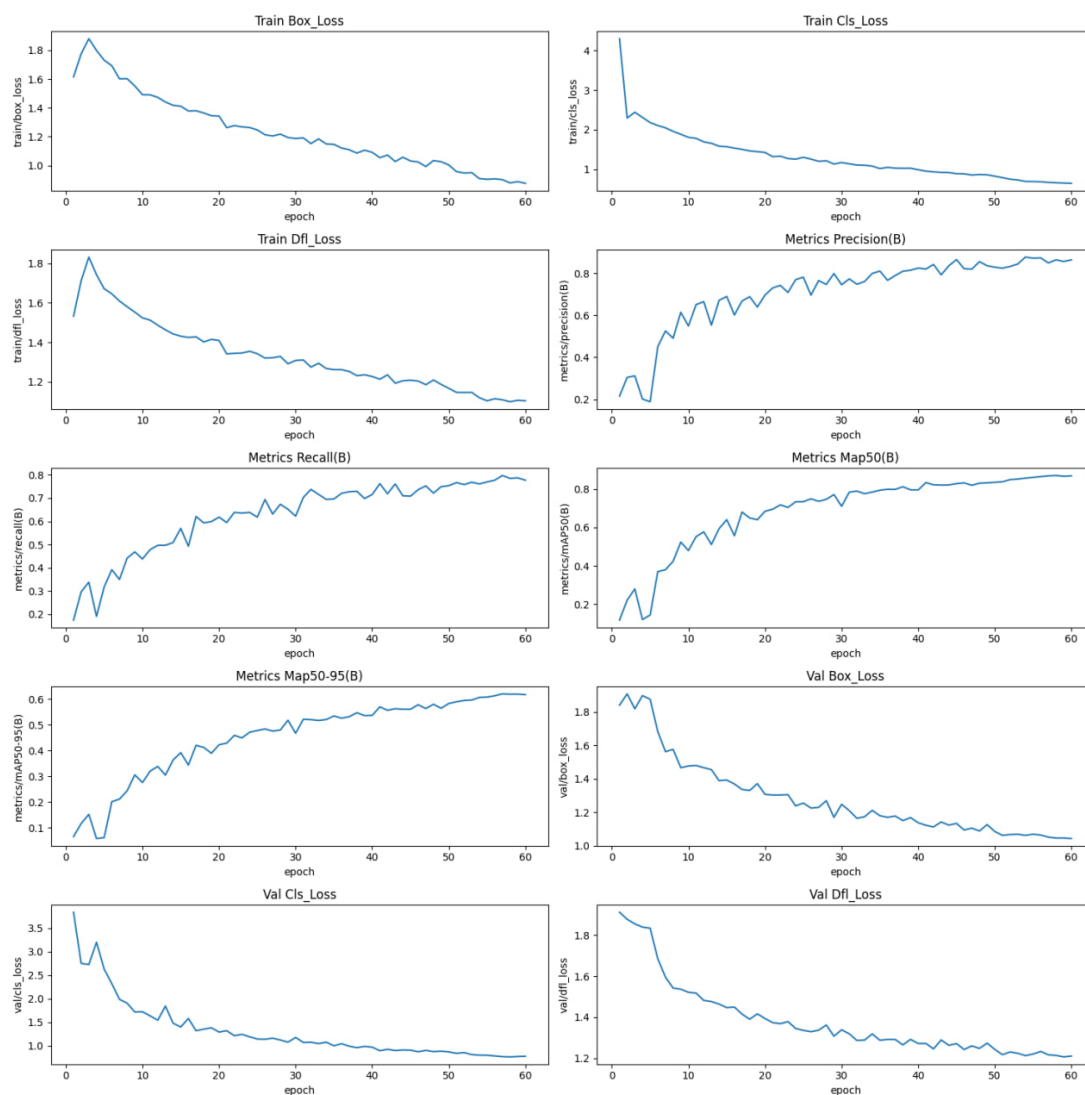


Рисунок 4.4.1 – Ключевые метрики обучения и валидации: потери (loss), точность (precision), полнота (recall), средняя точность (mAP).

4.4.2 Сравнительный анализ архитектур

Результаты эксперимента представлены в Таблице 1.

Таблица 1 - Сравнительные характеристики моделей YOLOv8

N	Модель	Аугментация	Время обучения (часы)	Время инференса (мс)	Точность (P)	Полнота (R)	mAP50	mAP50-95	F1
1	YOLOv8s	Нет	0.454	3.9	0.848	0.798	0.869	0.62	0.821
2	YOLOv8s	Да	0.630	3.8	0.835	0.79	0.846	0.602	0.810
3	YOLOv8m	Нет	0.883	9.4	0.844	0.754	0.844	0.594	0.796
4	YOLOv8m	Да	1.993	9.2	0.859	0.765	0.842	0.603	0.806
5	YOLOv8x	Нет	с10.293	42.0	0.883	0.713	0.825	0.59	0.790

Для расчёта F1-меры использовалась стандартная формула:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

Согласно теоретическим предпосылкам, увеличение размера и глубины нейронной сети должно способствовать улучшению качества детекции за счёт более мощной способности к извлечению признаков и обобщению. Предполагалось, что модели YOLOv8m и YOLOv8x продемонстрируют превосходство по основным метрикам (Precision, Recall, mAP) по сравнению с базовой моделью YOLOv8s, особенно при наличии аугментации данных.

Однако полученные результаты показали иную картину:

1) Снижение полноты с ростом размера модели

Модели YOLOv8m и YOLOv8x продемонстрировали более низкие значения Recall по сравнению с YOLOv8s. Например:

- YOLOv8s (без аугментации): Recall = 0.798;
- YOLOv8m (без аугментации): Recall = 0.754;
- YOLOv8x (без аугментации): Recall = 0.713.

Это может быть связано с недостаточным объёмом обучающей выборки. Более крупные модели имеют больше параметров и требуют большего количества данных для эффективного обучения. При ограниченном датасете возможны эффекты недообучения или переобучения на шум, что негативно влияет на обобщающую способность модели.

2) Отсутствие ожидаемого прироста mAP@50

Хотя ожидалось, что увеличение сложности модели приведёт к более точному детектированию объектов, результаты оказались противоположными:

- YOLOv8s: mAP50 = 0.869;
- YOLOv8m: mAP50 = 0.844;
- YOLOv8x: mAP50 = 0.825.

Это может указывать на то, что датасет не содержит большого числа сложных для детекции случаев (мелкие объекты, перекрытия), в которых сложные модели имели бы преимущество.

3) Рост вычислительных затрат без улучшения качества

Время инференса YOLOv8x составило 42.0 мс, что более чем в 10 раз превышает аналогичный показатель YOLOv8s (3.9 мс). Прирост времени обучения YOLOv8x составляет 2267% при этом метрики качества у YOLOv8x оказались ниже, что делает её использование неоправданным для данной задачи.

4.4.3 Влияние аугментации данных

Применение дополнительной аугментации не привело к значимому улучшению показателей. Это объясняется тем, что YOLOv8 уже содержит встроенные механизмы аугментации, которые активируются по умолчанию во время обучения. Аугментации применяются «на лету» (on-the-fly) к каждому батчу и включают:

- цветовые преобразования (изменения HSV),
- геометрические трансформации (масштабирование, сдвиг, отражение),
- Mosaic-аугментацию,
- AutoAugment (randaugment),
- случайное удаление частей изображения (erasing).

Таким образом, даже без явного увеличения датасета, модель получает достаточно разнообразных обучающих примеров.

Дополнительная аугментация оказала умеренно положительное влияние на модель YOLOv8m (Precision увеличилась с 0.844 до 0.859), но привела к снижению mAP у YOLOv8s (с 0.869 до 0.846). Это может свидетельствовать о внесении избыточного шума в небольшую исходную выборку, что ухудшает способность компактной модели к обобщению.

Также было зафиксировано ожидаемое увеличение времени обучения при включённой аугментации (например, для YOLOv8m — с 0.883 до 1.993 часа), в то время как время инференса практически не изменилось.

В целом, встроенные механизмы аугментации в YOLOv8 оказываются достаточными для обеспечения разнообразия обучающих примеров, а расширение выборки с помощью дополнительной аугментации может не привести к улучшению качества, что и произошло в нашем случае.

Эксперименты показали, что для задач детекции на малых датасетах (1470 изображений) оптимальной является модель **YOLOv8s**, демонстрирующая наилучший баланс точности ($P=0.848$), полноты ($R=0.798$) и скорости работы (3.9 мс). Увеличение сложности архитектуры (YOLOv8m/x) привело к снижению эффективности из-за недостаточного объема данных для их обучения. Дополнительная аугментация не дала значимого улучшения, так как встроенные механизмы YOLOv8 уже обеспечивают достаточное разнообразие данных.

Для дальнейшего повышения качества модели необходимо увеличить разнообразие исходных данных, проводя дополнительные съёмки сложных для детектирования сценариев.

4.5 Подбор гиперпараметров

С целью повышения точности детекции объектов на изображениях была проведена автоматическая оптимизация гиперпараметров с использованием

встроенной функции *tune()* библиотеки Ultralytics YOLOv8. В качестве базовой использовалась предобученная модель *yolov8s.pt*, а в качестве обучающего набора — изображения и аннотации, описанные в конфигурационном файле *data.yaml*.

Процесс тюнинга включал 30 итераций, каждая из которых состояла из 60 эпох обучения. Размер входных изображений был установлен равным 640 пикселям. Для оптимизации использовался стохастический градиентный спуск (SGD), а значение параметра случайности (*random seed*) фиксировано на уровне 42 для обеспечения воспроизводимости результатов. Результаты подбора гиперпараметров представлены на рисунке 4.5.1.

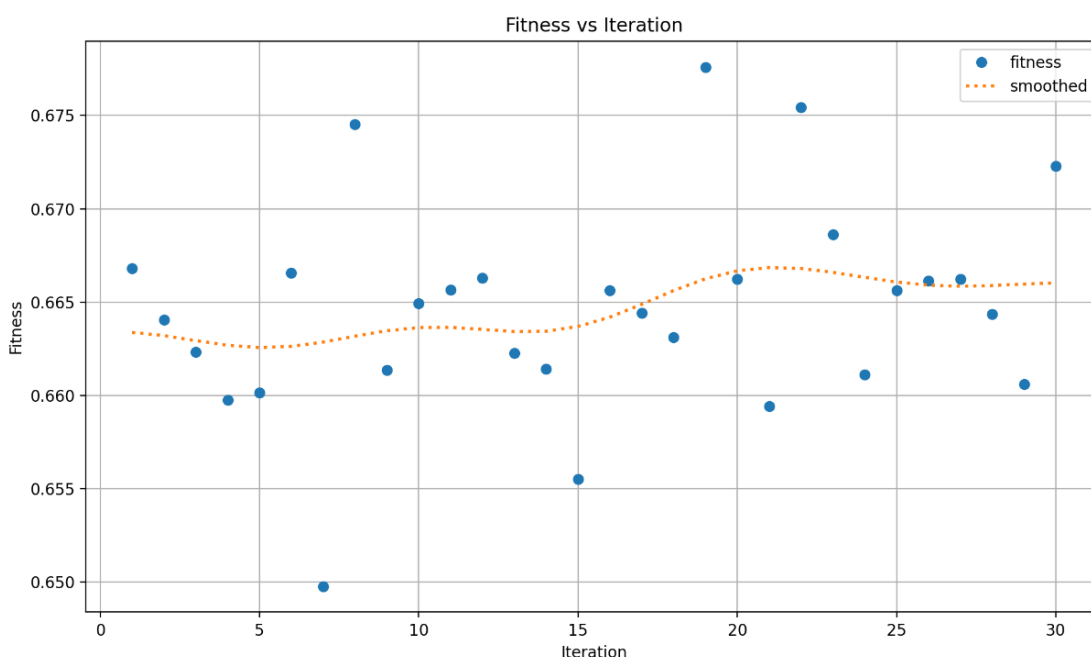


Рисунок 4.5.1 – Изменение метрики *fitness* в процессе подбора гиперпараметров

На графике показана динамика метрики *fitness* по мере выполнения итераций. Метрика *fitness* представляет собой агрегированную оценку качества модели, вычисляемую как взвешенное среднее нескольких показателей: точности (*Precision*), полноты (*Recall*), средних значений точности при порогах $\text{IoU} = 0.5$ ($\text{mAP}@0.5$) и в диапазоне от 0.5 до 0.95 ($\text{mAP}@0.5:0.95$). Наибольший вес в итоговом значении имеет $\text{mAP}@0.5$ (вес 0.9 по умолчанию), что делает его ключевым критерием при отборе лучших конфигураций.

Визуальный анализ графика позволяет отметить небольшой, но устойчивый рост метрики *fitness* до 20-й итерации, что свидетельствует о положительном влиянии оптимизации на итоговое качество модели.

В ходе тюнинга были автоматически определены следующие значения гиперпараметров:

- **обучение:** *lr0*: 0.01247, *lrf*: 0.00723, *momentum*: 0.8528, *weight_decay*: 0.00055, *warmup_epochs*: 3.50901, *warmup_momentum*: 0.83874;

- **функции потерь:** box: 6.13935, cls: 0.4425, dfl: 1.31937;
- **аугментации:** hsv_h: 0.01541, hsv_s: 0.84888, hsv_v: 0.41768, translate: 0.09992, scale: 0.53588, fliplr: 0.49988, mosaic: 0.96081;
- **отключённые аугментации** (установлены в 0): degrees, shear, perspective, flipud, mixup, copy_paste, bgr.

Для оценки эффективности подобранных гиперпараметров была проведена повторная процедура обучения модели на протяжении 100 эпох, с последующей валидацией на том же датасете. Результаты сравнения модели с параметрами по умолчанию и после оптимизации представлены в таблице 2.

Таблица 2 – Сравнение метрик до и после подбора гиперпараметров

Конфигурация	Precision (P)	Recall (R)	mAP0.5	mAP0.5:0.95	F1
Параметры по умолчанию	0.878	0.792	0.846	0.610	0.833
После тюнинга	0.858	0.840	0.874	0.635	0.849

Метрика **F1-score**, рассчитанная как гармоническое среднее между точностью и полнотой, увеличилась с 0.833 до 0.849. Это указывает на более сбалансированное поведение модели после тюнинга. Несмотря на незначительное снижение точности на 2 процента, наблюдается рост полноты (на 5%) и средних точностей при различных порогах IoU. Особенно важно улучшение mAP@0.5:0.95, что свидетельствует о повышении способности модели к точной локализации объектов при различных степенях перекрытия.

Таким образом, автоматический подбор гиперпараметров позволил достичь лучшего общего качества модели и повысить её способность к обобщению на валидационном наборе данных.

4.6 Адаптация модели для детектирования на Jetson Nano

Для обеспечения возможности работы модели детекции свалок твёрдых бытовых отходов на борту дрона с использованием одноплатного компьютера Jetson Nano потребовалась конвертация и оптимизация обученной модели YOLOv8. Данный процесс включает переход от исходного формата модели .pt (PyTorch) к формату .onnx, который более универсален и подходит для выполнения инференса с использованием аппаратного ускорения на устройствах с ограниченными ресурсами.

Формат .pt представляет собой сериализованное состояние модели PyTorch, включая архитектуру нейронной сети, обученные веса и параметры. Он подходит для обучения и инференса в среде PyTorch, но не обеспечивает достаточной производительности и совместимости для запуска модели вне этой

среды, особенно на встраиваемых устройствах. Для оптимизации модели был использован экспорт в формат Open Neural Network Exchange (ONNX), который представляет собой открытый формат для межплатформенной передачи нейросетевых моделей. ONNX позволяет выполнять инференс с помощью различных движков, включая ONNX Runtime с CUDA-ускорением, что критично для эффективной работы на Jetson Nano.

Экспорт модели выполняется с помощью встроенного метода библиотеки `ultralytics`, позволяющего преобразовать обученную модель в ONNX-формат с указанием параметров `opset=12`, `dynamic=True` и `simplify=True`, что обеспечивает совместимость, гибкость и снижение количества операций внутри графа:

```
from ultralytics import YOLO  
model = YOLO("path/to/best.pt")  
model.export(format="onnx", opset=12, dynamic=True, simplify=True)
```

После получения ONNX-модели производится загрузка модели с использованием библиотеки `onnxruntime`, при этом в качестве провайдеров вычислений указываются `'CUDAExecutionProvider'` и `'CPUExecutionProvider'`, что позволяет задействовать графический ускоритель Jetson Nano (GPU) при наличии поддержки.

Видеопоток с камеры обрабатывается в реальном времени с использованием библиотеки `OpenCV`. Каждый кадр предварительно подготавливается (изменение размера, нормализация, изменение порядка каналов), после чего передаётся в ONNX-модель для инференса. Выход модели проходит постобработку, включающую отбор объектов по порогу уверенности, преобразование координат ограничивающих рамок и применение немаксимального подавления для устранения дублирующихся предсказаний.

Если на кадре обнаружены объекты, соответствующие классу "свалка", результат визуализируется на изображении и сохраняется в отдельную директорию с временной меткой. Весь процесс реализован в цикле с возможностью досрочного завершения по нажатию клавиши.

Таким образом, предложенное решение позволяет выполнять детектирование свалок ТБО в реальном времени непосредственно на борту дрона с использованием Jetson Nano. Это даёт возможность снизить нагрузку на канал связи и повысить автономность системы мониторинга.

4.7 Визуальная и количественная оценка результатов

Для подтверждения качества работы модели была проведена визуальная оценка её предсказаний. На рисунках 3 и 4 представлены сравнительные результаты разметки данных: исходная разметка, созданная вручную (рисунок 3, `val_batch_labels`), и результаты предсказаний модели (рисунок 4, `val_batch_pred`).



Рисунок 4.7.1 – Исходная разметка, созданная вручную



Рисунок 4.7.2 – Результаты предсказаний модели

Количественное сравнение предсказаний модели с вручную размеченными данными показало, что из 38 объектов, присутствующих на изображениях, модель корректно идентифицировала 34, пропустив 4 объекта и допустив 9 ложных срабатываний. Доля правильно обнаруженных объектов составила 89,5%, что свидетельствует о высокой полноте модели.

Для оценки точности (precision) воспользуемся стандартной формулой:

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

где TP (true positives) - количество объектов, корректно обнаруженных моделью, FP (false positives) - число ложных срабатываний.

Подставляя полученные значения (TP=34, FP=9), вычисляем точность модели:

$$Precision = \frac{34}{34 + 9} = 0.79.$$

Таким образом, точность модели в данном примере составляет 79%, что демонстрирует её способность корректно идентифицировать целевые объекты при допустимом уровне ложных срабатываний.

Сравнение этих изображений демонстрирует, что модель уверенно детектирует объекты, корректно идентифицируя их границы. Совпадение большинства предсказанных рамок с размеченными вручную свидетельствует о высоком качестве работы модели не только в количественных метриках, но и с точки зрения визуального анализа. Это подтверждает, что модель способна справляться с реальными данными, содержащими разнообразные сцены и объекты различного масштаба. Такие результаты доказывают, что выбранная архитектура модели и параметры обучения обеспечивают высокую точность и надёжность детекции.

4.8 Анализ ложных срабатываний модели и их причины

Несмотря на высокие количественные показатели эффективности модели, визуальная оценка выявила ряд случаев ошибочной детекции. Анализ ложных срабатываний позволяет выявить уязвимости алгоритма и определить направления для дальнейшего улучшения системы.

На изображении 4.8.1 зафиксировано ложное срабатывание модели: область крыши здания была ошибочно классифицирована как скопление отходов с уровнем уверенности 0,75. Данная ошибка обусловлена схожестью визуальных признаков объекта фона и целевого класса, такими как беспорядочная текстура,

наличие контрастных элементов и цветовые особенности покрытия крыши, напоминающие распределённые мусорные кучи.



Рисунок 4.8.1 – Ложное срабатывание модели: детекция фрагмента крыши как скопления ТБО

Дополнительно выявлены случаи некорректной работы модели на изображениях, представленных на рисунках 4.8.2 и 4.8.3. На первом из них модель ошибочно детектировала участки снега вблизи деревьев как объекты класса «dump». Вероятно, причиной стало сходство текстуры и цвета между фрагментами природных объектов и реальными мусорными скоплениями, представленными в обучающем наборе данных.

На следующем рисунке аналогичная ошибка возникла на участках снега, частично загрязнённого почвой. Модель зафиксировала область, содержащую пятна грязи, как мусор, в то время как сама мусорная куча осталась незамеченной. Это указывает на недостаточную дифференциацию модели между мусором и естественными загрязнёнными участками почвы или снегового покрова.



Рисунок 4.8.2 – Ошибочная классификация скопления снега как мусора



Рисунок 4.8.3 – Ложноположительная детекция загрязнённого снежного покрова и ложноотрицательная детекция мусора

Основной причиной этих ошибок являлась недостаточная репрезентативность обучающей выборки. Поскольку сбор данных проводился преимущественно в бесснежный период, модель не имела достаточного количества примеров для корректного различения снега, грязи и истинных скоплений отходов. Сбор дополнительных данных и дообучение модели

помогли избавиться от ошибок, связанных со снегом. Корректную детекцию этой же территории можно увидеть на рисунке 4.8.4.

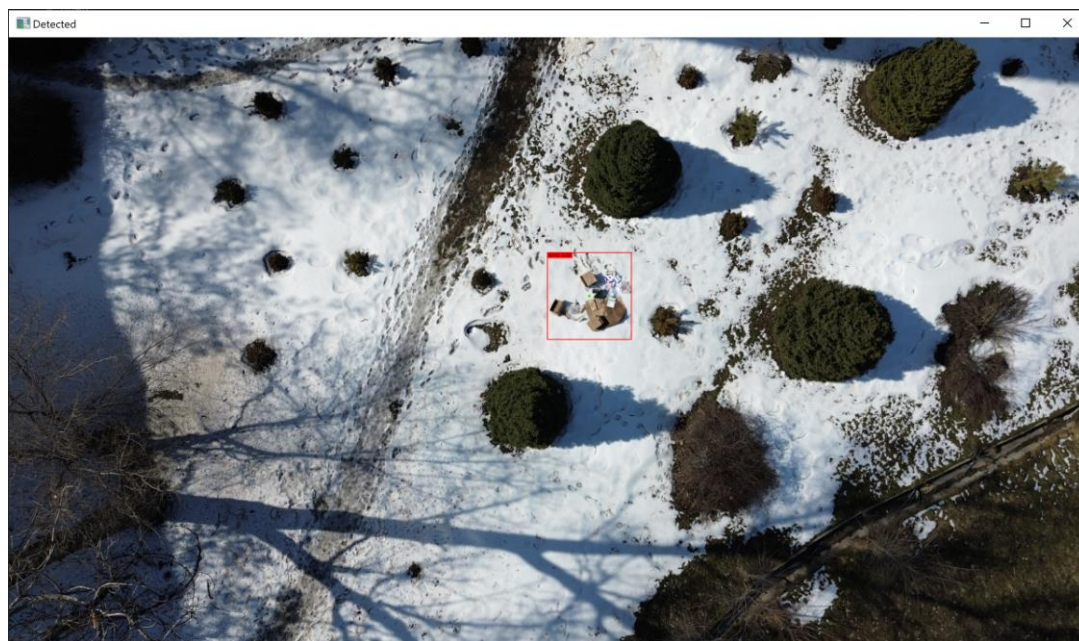


Рисунок 4.8.4 – Корректная детекция участка, который вызывал ошибки в более ранней версии

Эти примеры подтверждают необходимость увеличения разнообразия тренировочного датасета за счёт включения изображений с похожими на мусор элементами природной среды (грязь, снег, камни, обнажённая почва). Это позволит улучшить способность модели различать мусорные скопления и естественные объекты, минимизируя количество ложных срабатываний.

4.9 Определение координат объектов на снимках с БПЛА

Одной из важных задач в рамках построения автоматизированной системы мониторинга ТБО с применением беспилотных летательных аппаратов БПЛА является определение географических координат обнаруженных объектов на аэрофотоснимках. Данная информация необходима для последующей локализации свалок на карте и организации выездных проверок.

Для определения координат объектов, распознанных моделью YOLOv8 на изображении, был реализован скрипт, включающий следующие этапы:

1) Извлечение метаданных изображения

Для каждого снимка, полученного с дрона, с помощью внешней утилиты ExifTool извлекаются метаданные EXIF, содержащие GPS-координаты местоположения БПЛА в момент съёмки (широта, долгота) и относительную высоту над уровнем земли (Relative Altitude). Координаты представлены в

формате DMS (градусы, минуты, секунды) и преобразуются в десятичную систему с использованием собственной функции, учитывающей направление (север/юг, восток/запад).

2) Детектирование объектов

Изображения обрабатывается с помощью предварительно обученной модели YOLOv8, которая возвращает список обнаруженных объектов в виде прямоугольных ограничивающих рамок, а также предсказанный класс со соответствующей степенью уверенности. Для каждого объекта вычисляется центр рамки в пиксельных координатах.

3) Геопривязка обнаруженных объектов

Рассчитанные пиксельные координаты центра объекта сопоставляются с географической системой координат, исходя из известных параметров съёмки. В рамках данной реализации принято приближение, согласно которому камера дрона направлена строго вертикально (угол наклона 90°), а линза имеет фиксированный угол обзора по диагонали.

Для дрона DJI Mini 2, использованного в эксперименте, угол обзора составляет примерно 84° . При известной высоте полёта h , это позволяет оценить ширину участка местности, попавшей в кадр, по формуле:

$$W = 2h \cdot \tan \frac{\theta}{2},$$

где θ — угол обзора камеры. Далее определяется масштаб изображения в метрах на пиксель:

$$S = \frac{W}{w},$$

где w — ширина изображения в пикселях.

Смещения центра объекта относительно центра изображения в пикселях ($\Delta x, \Delta y$) преобразуются в метры, а затем — в разницу географических координат:

$$\Delta\phi = \frac{\Delta y \cdot S}{111000},$$
$$\Delta\lambda = \frac{\Delta x \cdot S}{111000 \cdot \cos(\phi)},$$

где ϕ — широта точки съёмки.

Таким образом, итоговые координаты объекта определяются как:

$$\phi(\text{объекта}) = \phi(\text{дрона}) - \Delta\phi,$$

$$\lambda(\text{объекта}) = \lambda(\text{дрона}) - \Delta\lambda.$$

Знак минус в выражении для широты объясняется особенностью системы координат изображения: ось Y направлена вниз, в отличие от географической системы.

Для наглядности на исходное изображение автоматически наносятся прямоугольные рамки вокруг распознанных объектов, подписываются их классы и степени уверенности, а также выделяются центры каждого объекта (красные точки на рисунке 4.9.3), для которых производятся расчёты координат. Это обеспечивает удобную верификацию результатов и визуальную интерпретацию детекции.

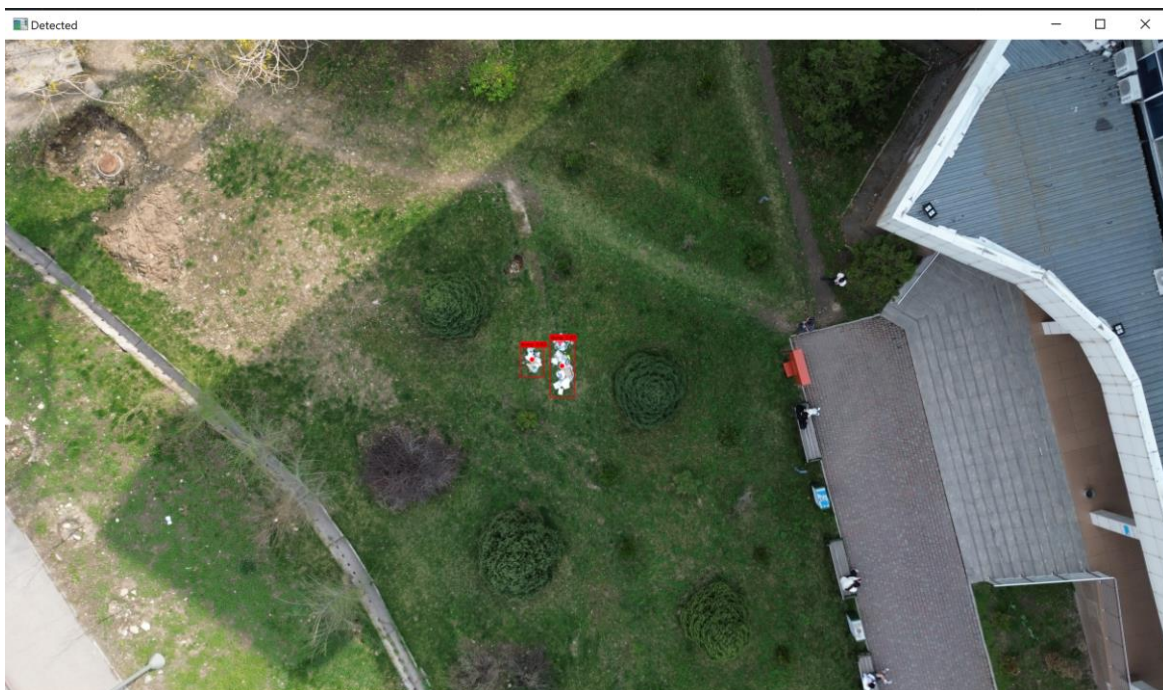


Рисунок 4.9.1 – Детекция мусора, с отмеченными центрами

Вывод программы для данного изображения:

*[INFO] Координаты дрона: 43.236875, 76.9351638888889, Высота: 30.6 м
image 1/1 C:\Users\Ludmila\Desktop_\\10.04.2025 \DJI_0134.JPG: 384x640 2
dumps, 51.1ms
Speed: 3.0ms preprocess, 51.1ms inference, 157.0ms postprocess per image at shape
(1, 3, 384, 640)
[INFO] Найдено 2 объекта
- Свалка: Центр рамки (x: 1915, y: 1120)
Геокоординаты свалки: lat = 43.236876, lon = 76.935152
- Свалка: Центр рамки (x: 1812, y: 1098)
Геокоординаты свалки: lat = 43.236878, lon = 76.935138
Размер изображения: 4000 x 2250*

Разработанный алгоритм позволяет с достаточной точностью определять координаты объектов на аэрофотоснимках, полученных с БПЛА, при наличии базовых параметров съёмки. Полученные координаты могут быть использованы для построения карт замусоренных участков, их систематизации и интеграции в геоинформационные системы. Допущения о вертикальной ориентации камеры и постоянстве угла обзора оправданы в контексте задач предварительного анализа и экологического мониторинга. Пилот дрона старался фиксировать угол наклона и на большинстве снимках он действительно равен 90° или на несколько градусов отклоняется от этого значения.

ЗАКЛЮЧЕНИЕ

В рамках данного исследования была разработана и протестирована система распознавания ТБО на основе архитектуры YOLOv8, предназначенная для работы с изображениями, полученными с БПЛА. Разработанный метод включает последовательные этапы сбора данных, их подготовки, обучения модели и оптимизации гиперпараметров.

Анализ количественных метрик, таких как precision (84.8%), recall (81.5%), и mAP@0.5 (85.1%), подтверждает достаточную для практики эффективность предложенного метода детекции ТБО. Дополнительно, визуальная оценка предсказаний модели демонстрирует её способность успешно идентифицировать целевые объекты с минимальным количеством ошибок. Это доказывает, что модель обладает высокой обобщающей способностью и устойчивостью к различным условиям съёмки.

Однако, несмотря на достигнутые результаты, остаются области для дальнейшего совершенствования. Например, метрика mAP@0.5:0.95 (56.3%) указывает на возможность повышения точности модели при более строгих порогах IoU. В будущем планируется увеличить объём данных за счёт добавления изображений различных типов отходов и условий съёмки, а также провести дополнительные эксперименты с альтернативными архитектурами моделей.

Разработанный подход обладает потенциалом для использования в реальных задачах экологического мониторинга, включая автоматизацию процессов идентификации и контроля свалок ТБО, что может способствовать повышению эффективности управления отходами и снижению их негативного воздействия на окружающую среду.

В таблице 3 приведены принятые в тексте сокращения и их расшифровки.

Таблица 3 – Принятые в тексте сокращения

Сокращение	Расшифровка
ТБО	Твёрдые бытовые отходы
БПЛА	Беспилотный летательный аппарат
ГИС	Геоинформационная система
ИИ	Искусственный интеллект
YOLO	You Only Look Once
CNN	Convolutional neural network
FPS	Frames Per Second
IoU	Intersection over Union
GPS	Global Positioning System
mAP	Mean Average Precision
DMS	Degrees, minutes, seconds
TP	True positives
FP	False positives
ONNX	Open Neural Network Exchange
P	Precision
R	Recall
FN	False Negative
HSV	Hue, Saturation, Value

СПИСОК ЛИТЕРАТУРЫ

1. Hoornweg D., Bhada-Tata P. What a waste: A global review of solid waste management. World Bank, 2012.
2. Калиева С.С., Төлепов Ә.Е., Хойч А., Шаяхметова Л.М. Экономическая оценка политики по сокращению ТБО в Республике Казахстан с использованием метода условной оценки (CV). Вестник университета «Туран». – 2024. – № 3. – С. 23-34. <https://doi.org/10.46914/1562-2959-2024-1-3-23-34>.
3. Холявко Т.И. Динамика количества твердых бытовых отходов и их морфологический состав на примере Санкт-Петербурга, Стокгольма, Токио и Хельсинки Приоритетные научные направления: от теории к практике. – 2014. – №10. – С. 16-21.
4. Сатбаева А. Около 80 миллиардов iPhone. Сколько мусора накопилось в Алматы и Астане//Tengrinews.kz. – 2023 [Электронный ресурс]. – URL: https://tengrinews.kz/kazakhstan_news/okolo-80-milliardov-iphone-skolko-musora-nakopilos-almaty-i-493313/ (дата обращения: 28.04.2025).
5. Бекежанов Д. Проблемы и перспективы утилизации твердых бытовых отходов в Республике Казахстан // Вестник КазНУ. – 2017. – № 1(81). – С. 227- 233.
6. Булекова А.А., Габдуллина С. Проблема утилизации ТБО // Вестник ЗКУ. – 2021. – № 1(81).
7. Радкевич М.В., Мухаммадалиева Ф. Проблема несанкционированных свалок в Республике Узбекистан // Universum: химия и биология. Электрон. научн. журн. 2022. № 4(94). URL: <https://7universum.com/ru/nature/archive/item/13341> (дата обращения: 22.03.2025).
8. Скрипка Г.И., Сапрыгин В.В., Ивлиева О.В., Беспалова Л.А., Глинка В.В. Опыт использования беспилотных летательных аппаратов для выявления мест размещения отходов в водоохранных зонах // Известия высших учебных заведений. Северо-Кавказский регион. Естественные науки. – 2023. – № 3(219). – С. 131-139. doi: 10.18522/1026-2237-2023-3-131-139.
9. Семенова М.В., Архипова О.Е. Современные методы мониторинга экологического состояния территории // Геоинформационные технологии и космический мониторинг. 2019. Т. 2, № 4. С. 92–97. DOI: 10.23885/2500-123X-2019-2-4-92-97
10. Albrekht, V.; Mukhamediev, R.I.; Popova, Y.; Muhamedijeva, E.; Botaibekov, A. Top2Vec Topic Modeling to Analyze the Dynamics of Publication Activity Related to Environmental Monitoring Using Unmanned Aerial Vehicles. Publications 2025, 13, 15. <https://doi.org/10.3390/publications13020015>
11. E. Cano, R. Horton, C. Liljegren, and M.D. Bulanon. Comparison of small unmanned aerial vehicle performance using image processing. Journal of Imaging 3(1):4, January 2017. MDPI, Basel, Switzerland. [Online]. Available: <https://doi.org/10.3390/jimaging3010004>

12. Мониторинг мусорных полигонов с применением дронов // Aeromotus. [Электронный ресурс]. – URL: <https://aeromotus.ru/monitoring-of-landfills-from-uavs/> (дата обращения: 11.01.2025).
13. Летающие дроны и умные камеры: как в Нур-Султане борются с несанкционированными свалками // Kazlenta.kz. – 2021. – 7 дек. – URL: <https://kazlenta.kz/44420-letayuschie-drony-i-umnye-kamery-kak-v-nur-sultane-boryutsya-s-nesankcionirovannymi-svalkami.html> (дата обращения: 11.01.2025).
14. Casas E., Ramos L., Romero C., Rivas-Echeverría F. A review of computer vision applications for asset inspection in the oil and gas Industry // Journal of Pipeline Science and Engineering. – 2024. – P. 100246. <https://doi.org/10.1016/j.jpse.2024.100246>.
15. Gheorghe C., Duguleana M., Boboc R.G., Postelnicu C.C. Analyzing Real-Time Object Detection with YOLO Algorithm in Automotive Applications: A Review // CMES - Computer Modeling in Engineering and Sciences. – 2024. – Vol. 141, Issue 3. – P. 1939-1981. <https://doi.org/10.32604/cmes.2024.054735>.
16. Explanation of all of YOLO series Part 11. [Электронный ресурс]. – URL: https://zenn.dev/yuto_mo/articles/14a87a0db17dfa (дата обращения: 11.01.2025).
17. Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. – P. 779–788.
18. Русаков К.Д. Использование глубокого обучения и беспилотных летательных аппаратов для обнаружения мусора на водной поверхности // Всероссийское совещание управления. – 2024. – С. 1157–1161.
19. Han W., Luo W., Jin Y., et al. A Deep Learning Model for Automatic Plastic Waste Monitoring Using Unmanned Aerial Vehicle (UAV) Data // ESS Open Archive. – 2021. <https://doi.org/1010.1002/essoar.10507932.1>.
20. Тимошкин М.С., Миронов А.Н., Леонтьев А.С. Сравнение YOLO V5 и Faster R-CNN для обнаружения людей на изображении в потоковом режиме // Международный научно-исследовательский журнал. – 2022. – № 6 (120). – Ч. 1. – DOI: 10.23670/IRJ.2022.120.6.020.
21. Liao, Y.-H.; Juang, J.-G. Real-Time UAV Trash Monitoring System. Appl. Sci. – 2022. – 12. – P. 1838. <https://doi.org/10.3390/app12041838>
22. Казаков С., Азаренко Н. Интеллектуальная система обнаружения и учета мусора на территории муниципалитета // Сборник статей IV Всероссийской научно-практической конференции, приуроченной к Году науки и технологий в России «Вызовы цифровой экономики: тренды развития в условиях последствий пандемии COVID-19». – Брянск, 2021. – С. 10–12.
23. Джозеф Нельсон. (5 января 2024 г.). Как маркировать данные изображений для моделей компьютерного зрения. Блог Roboflow: <https://blog.roboflow.com/tips-for-how-to-label-images/>

24. Jocher, G., Munawar, M. R., & Vina, A. (2025) Глубокое погружение в метрики производительности [Электронный ресурс]. URL: <https://docs.ultralytics.com/ru/guides/yolo-performance-metrics/#object-detection-metrics> (дата обращения: 11.05.2025)
25. Набор данных COCO. [Электронный ресурс]. – URL: <https://docs.ultralytics.com/ru/datasets/detect/coco/> (дата обращения: 11.01.2025).

Приложение А

Техническое задание

А.1.5 Техническое задание на Распознавание твёрдых бытовых отходов в ходе мониторинга городской агломерации с помощью БПЛА

Настоящее техническое задание определяет требования к разработке алгоритма детектирования несанкционированных свалок твёрдых бытовых отходов на снимках, полученных с беспилотных летательных аппаратов. Данный алгоритм предназначен для автоматизированного экологического мониторинга, что позволит муниципальным службам оперативно выявлять нарушения и предпринимать меры по устранению свалок. Это решение снижает необходимость привлечения значительных человеческих ресурсов для обследования больших территорий, включая труднодоступные участки.

А.1.5.1 Основание для разработки

Алгоритм разрабатывается на основе согласия научного руководителя на выбранную дипломником тему и необходимости выполнения работ по программе BR21881908 «Комплекс экологического сопровождения городской агломерации».

А.1.5.2 Назначение

Разрабатываемое программное обеспечение (ПО) предназначено для автоматического выявления скоплений твёрдых бытовых отходов на снимках, полученных с борта беспилотных летательных аппаратов. ПО будет использоваться для мониторинга городской агломерации, анализа динамики загрязнения территорий и повышения эффективности экологического контроля.

А.1.5.3 Требования к функциональным характеристикам

Программный инструмент должен обеспечивать возможность выполнения следующих функций:

1. Обнаружение участков, содержащих несанкционированные свалки ТБО;
2. Вывод координат зон, на которых обнаружены свалки;
3. Вывод координат обнаруженных свалок для отображения на портале (разрабатываемого в рамках BR21881908).

А.1.5.4 Требования к надежности

Система должна обеспечивать:

- Высокую точность детектирования объектов на изображениях (не менее 75%);
- Стабильную работу при обработке изображений различных разрешений и форматов;
- Устойчивость к изменению условий съёмки (локация, освещение, погодные условия).

Приложение Б

Листинг (код) программы

Скрипт для подбора гиперпараметров:

```
import os
import torch
from ultralytics import YOLO
def tune_hyperparameters(image_size, project_dir):
    """Подбор гиперпараметров для YOLOv8."""
    # Загружаем предобученную модель YOLOv8s
    model = YOLO('yolov8s.pt')
    print("Начинается подбор гиперпараметров...")
    # Запуск подбора гиперпараметров
    model.tune(
        data=r"C:\Users\Ludmila\Desktop\dump\v2\data.yaml",
        epochs=60,
        imgsz=image_size,
        seed= 42,
        iterations=30,
        optimizer='SGD',
        val=True,
        plots=True,
        project=project_dir,
        name="v2_hyp3_yolo8s"
    )
    # Путь к лучшему hyp.yaml
    best_hyp_path = os.path.join(project_dir, "v2_hyp3_yolo8s", "best_hyperparameters.yaml")
    print("\n Подбор завершён. Лучшие гиперпараметры:")
    if os.path.exists(best_hyp_path):
        with open(best_hyp_path, "r") as f:
            print(f.read())
    else:
        print("Файл hyp.yaml не найден")
if __name__ == '__main__':
    torch.multiprocessing.freeze_support()
    # Папка, в которую сохраняются результаты подбора
    project_dir = r"C:\Users\Ludmila\Desktop\dump\YOLO_outputs_2"
    image_size = 640
    # Запуск подбора гиперпараметров
    tune_hyperparameters(image_size, project_dir)
```

Скрипт для обучения модели с использованием подобранных гиперпараметров, а также для построения графиков потерь и точности на обучении и валидации:

```
import os
import random
import cv2
import pandas as pd
import matplotlib.pyplot as plt
```

```

import seaborn as sns
import torch
from ultralytics import YOLO
import yaml
def train_model(image_size, project_dir):
    """Функция для обучения модели YOLO с заданными гиперпараметрами."""
    # Загружаем словарь с гиперпараметрами из yaml-файла
    with
open(r"C:\Users\Ludmila\Desktop\dump\YOLO_outputs_2\v2_hyp3_yolo8s2\best_hyperparameter
s.yaml", "r") as f:
    hyp_dict = yaml.safe_load(f)
    # Загружаем предобученную модель YOLOv8s
    model = YOLO('yolov8s.pt')
    model.train(
        data=r"C:\Users\Ludmila\Desktop\dump\v2\data.yaml",
        epochs= 150,
        imgsz=image_size,
        seed=42,
        project=project_dir,
        name="v2_yolo8s_hyp3_150",
        **hyp_dict
    )
def plot_training_metrics(csv_path):
    """Построение графиков метрик из тренировочного процесса."""
    # Загружаем CSV-файл с результатами обучения
    df = pd.read_csv(csv_path)
    # Убираем лишние пробелы из названий колонок, если они есть
    df.columns = df.columns.str.strip()
    # Создаем 5 строк по 2 графика (итого 10 графиков)
    fig, axs = plt.subplots(nrows=5, ncols=2, figsize=(15, 15))
    # Строим графики для каждой метрики
    sns.lineplot(x='epoch', y='train/box_loss', data=df, ax=axs[0, 0])
    sns.lineplot(x='epoch', y='train/cls_loss', data=df, ax=axs[0, 1])
    sns.lineplot(x='epoch', y='train/dfl_loss', data=df, ax=axs[1, 0])
    sns.lineplot(x='epoch', y='metrics/precision(B)', data=df, ax=axs[1, 1])
    sns.lineplot(x='epoch', y='metrics/recall(B)', data=df, ax=axs[2, 0])
    sns.lineplot(x='epoch', y='metrics/mAP50(B)', data=df, ax=axs[2, 1])
    sns.lineplot(x='epoch', y='metrics/mAP50-95(B)', data=df, ax=axs[3, 0])
    sns.lineplot(x='epoch', y='val/box_loss', data=df, ax=axs[3, 1])
    sns.lineplot(x='epoch', y='val/cls_loss', data=df, ax=axs[4, 0])
    sns.lineplot(x='epoch', y='val/dfl_loss', data=df, ax=axs[4, 1])
    # Добавляем заголовки на графики на основе названия метрик
    for ax in axs.flat:
        ax.set_title(ax.get_ylabel().replace("/", " ").title())
    plt.tight_layout()
    plt.show()
if __name__ == '__main__':
    # Для корректной работы multiprocessing на Windows
    torch.multiprocessing.freeze_support()
    # Задаем директорию, куда сохраняются результаты обучения
    project_dir = r"C:\Users\Ludmila\Desktop\dump\YOLO_outputs"

```

```

# Тренировка модели
height = 640
train_model(height, project_dir)
# Путь к файлу с результатами обучения
results_csv = os.path.join(project_dir, "v2_yolo8s_hyp3_150", "results.csv")
# Строим графики по метрикам обучения
plot_training_metrics(results_csv)

```

Скрипт для детектирования свалок ТБО на сохранённых изображениях:

```

import cv2
import os
import matplotlib.pyplot as plt
from ultralytics import YOLO
# Загружаем модель
model =
YOLO(r"C:\Users\Ludmila\Desktop\dump\YOLO_outputs\v2_yolo8s_hyp\weights\best.onnx")
# Папка с изображениями
input_dir = r"C:\Users\Ludmila\Desktop\Все_датасеты\ТБО\11.03.2025 Мусор возле
ККУЦ\фото с дрона"
output_dir = r"C:\Users\Ludmila\Desktop\Все_датасеты\ТБО\11.03.2025 Мусор возле
ККУЦ\annotated"
# Создаём папку для сохранения, если её нет
os.makedirs(output_dir, exist_ok=True)
# Получаем список всех файлов в папке
image_files = [f for f in os.listdir(input_dir) if f.lower().endswith((".jpg", ".jpeg", ".png"))]
# Обрабатываем все изображения в папке
for image_file in image_files:
    image_path = os.path.join(input_dir, image_file)
    output_path = os.path.join(output_dir, f"annotated_{image_file}")
    print(f"Обрабатываем: {image_path}")
    # Запускаем предсказание
    results = model.predict(source=image_path, imgsz=640)
    for result in results:
        # Получаем размеченное изображение
        annotated_image = result.plot()
        annotated_image = cv2.cvtColor(annotated_image, cv2.COLOR_RGB2BGR)
        # Сохраняем изображение
        cv2.imwrite(output_path, annotated_image)
        print(f"Сохранено: {output_path}")
        # Отображаем изображение через Matplotlib (не закрывая предыдущие)
        plt.figure(figsize=(6, 6))
        plt.imshow(cv2.cvtColor(annotated_image, cv2.COLOR_BGR2RGB))
        plt.axis('off')
        plt.title(f"Detections: {image_file}")
        plt.show(block=False) # Не блокирует выполнение программы
print("Обработка завершена.")

```

Скрипт для детектирования свалок ТБО в режиме реального времени с использованием web-камеры:

```
import cv2
import os
import numpy as np
from datetime import datetime
import onnxruntime as ort # для запуска модели в формате ONNX
# Путь к модели
onnx_path =
r"C:\Users\Ludmila\Desktop\dump\YOLO_outputs\v2_yolo8s_hyp\weights\best.onnx"
session = ort.InferenceSession(onnx_path, providers=['CPUExecutionProvider'])
input_name = session.get_inputs()[0].name
output_name = session.get_outputs()[0].name
# Папка, куда будем сохранять кадры с обнаруженными объектами
output_dir = r"C:\Users\Ludmila\Desktop\dump\detected_frames_jetson"
os.makedirs(output_dir, exist_ok=True)
# Инициализируем видеопоток с камеры
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("Ошибка: не удалось открыть камеру")
    exit()
def preprocess(frame):
    """
    Предобработка кадра: изменение размера, нормализация, изменение порядка каналов.
    """
    img = cv2.resize(frame, (640, 640)) # Приводим кадр к нужному размеру
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Конвертируем BGR → RGB
    img = img.astype(np.float32) / 255.0 # Нормализация значений [0,1]
    img = np.transpose(img, (2, 0, 1)) # Изменяет форму массива с (высота, ширина, каналы) →
    (каналы, высота, ширина)
    img = np.expand_dims(img, axis=0) #Добавляет новую размерность спереди — размерность
    батча. (Изменяет форму массива с (3, 640, 640) → (1, 3, 640, 640))
    return img
def postprocess(outputs, orig_shape, conf_thres=0.3, iou_thres=0.4):
    """
    Постобработка результатов модели:
    - отбор по порогу уверенности
    - преобразование координат
    - non-maximum suppression (NMS)
    outputs — результат работы модели (массив numpy)
    orig_shape — исходное разрешение кадра, чтобы потом правильно масштабировать
    координаты
    conf_thres — порог уверенности (если объект предсказан слишком неуверенно, мы его
    игнорируем)
    iou_thres — порог перекрытия (IoU) для подавления похожих боксов (используется в NMS)
    """
    predictions = outputs[0] # shape: (1, 5, 8400)
    # Убираем размерность батча
```

```

predictions = np.squeeze(predictions, axis=0) # shape: (5, 8400)
# Транспонируем массив, чтобы получить по одной строке на каждый предсказанный
объект: одна строка = одна детекция
predictions = predictions.transpose(1, 0) # shape: (8400, 5)
# Отделяем координаты и confidence
boxes = predictions[:, :4] # cx, cy, w, h
confidences = predictions[:, 4] # уверенность
# Фильтруем по порогу уверенности
mask = confidences > conf_thres
boxes = boxes[mask]
confidences = confidences[mask]
if len(boxes) == 0:
    return []
# Преобразуем координаты из формата YOLO (cx, cy, w, h) в (x1, y1, x2, y2)
boxes_xyxy = np.zeros_like(boxes)
boxes_xyxy[:, 0] = boxes[:, 0] - boxes[:, 2] / 2
boxes_xyxy[:, 1] = boxes[:, 1] - boxes[:, 3] / 2
boxes_xyxy[:, 2] = boxes[:, 0] + boxes[:, 2] / 2
boxes_xyxy[:, 3] = boxes[:, 1] + boxes[:, 3] / 2
# Масштабируем координаты под исходное разрешение кадра
h, w = orig_shape[:2]
scale_x, scale_y = w / 640, h / 640
boxes_xyxy[:, [0, 2]] *= scale_x
boxes_xyxy[:, [1, 3]] *= scale_y
# Применяем non-maximum suppression (NMS)
indices = cv2.dnn.NMSBoxes(
    bboxes=boxes_xyxy.tolist(),
    scores=confidences.tolist(),
    score_threshold=conf_thres,
    nms_threshold=iou_thres
)
result = []
if len(indices) > 0:
    for i in indices.flatten():
        result.append((boxes_xyxy[i], confidences[i], 0)) # всегда класс 0
return result
class_names = ["dump"]
while True:
    ret, frame = cap.read()
    if not ret:
        break
    # Предобрабатываем кадр и передаём в модель
    input_tensor = preprocess(frame)
    outputs = session.run([output_name], {input_name: input_tensor})
    detections = postprocess(outputs, frame.shape)
    # Перебираем все обнаруженные прямоугольники на кадре
    for box, score, cls_id in detections:
        x1, y1, x2, y2 = map(int, box)

```

```

label = f'{class_names[cls_id]} {score:.2f}'
# Отображаем прямоугольник и метку
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
cv2.putText(frame, label, (x1, y1 - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1)
# Сохраняем кадр, если есть детекции
if len(detections) > 0:
    timestamp = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
    path = os.path.join(output_dir, f'detected_{timestamp}.jpg')
    cv2.imwrite(path, frame)
    print(f'Сохранён кадр: {path}')
# Показываем окно с детекцией в реальном времени
cv2.imshow("Live Detection", frame)
# Останавливаем по нажатию 'q' или 'й'
key = cv2.waitKey(1) & 0xFF
if key in [ord('q'), ord('й')]:
    break
# Освобождаем ресурсы
cap.release()
cv2.destroyAllWindows()

```

Скрипт для разделение исходного датасета на обучающую, валидационную, тестовую выборки и для создания структуры папок, удобной для дальнейшей работы:

```

import os
import shutil
import random
# Папка с исходными изображениями
image_dir = r"C:\Users\Ludmila\Desktop\dump\v2_all\images"
# Папка с соответствующими аннотациями (YOLO-формат: *.txt)
label_dir = r"C:\Users\Ludmila\Desktop\dump\v2_all\labels"
# Базовая директория для нового разбиения датасета
dataset_base = r"C:\Users\Ludmila\Desktop\dump\v2_div"
# Задаём структуру новых папок для train/val/test
split_dirs = {
    "train": {
        "images": os.path.join(dataset_base, "train", "images"),
        "labels": os.path.join(dataset_base, "train", "labels")
    },
    "val": {
        "images": os.path.join(dataset_base, "val", "images"),
        "labels": os.path.join(dataset_base, "val", "labels")
    },
    "test": {
        "images": os.path.join(dataset_base, "test", "images"),
        "labels": os.path.join(dataset_base, "test", "labels")
    },
}

```

```

# Создаём папки, если они ещё не существуют
for split in split_dirs.values():
    for path in split.values():
        os.makedirs(path, exist_ok=True)
# Получаем список всех файлов изображений с нужными расширениями
image_files = [f for f in os.listdir(image_dir) if f.endswith(('.jpg', '.png', '.jpeg'))]
# Перемешиваем список изображений для случайного разбиения
random.shuffle(image_files)
# Считаем общее количество изображений
num_total = len(image_files)
# Определяем количество файлов для train/val/test
n_train = int(num_total * 0.7) # 70% — обучение
n_val = int(num_total * 0.2) # 20% — валидация
n_test = num_total - n_train - n_val # оставшиеся 10% — тест
# Формируем словарь с разделёнными наборами данных
dataset_splits = {
    "train": image_files[:n_train],
    "val": image_files[n_train:n_train + n_val],
    "test": image_files[n_train + n_val:]
}
# Функция для перемещения изображений и соответствующих аннотаций
def move_files(file_list, split_name):
    for file_name in file_list:
        # Полный путь к изображению
        src_image = os.path.join(image_dir, file_name)
        dst_image = os.path.join(split_dirs[split_name]["images"], file_name)
        # Формируем имя и путь к аннотации, заменив расширение на .txt
        label_name = os.path.splitext(file_name)[0] + ".txt"
        src_label = os.path.join(label_dir, label_name)
        dst_label = os.path.join(split_dirs[split_name]["labels"], label_name)
        # Перемещаем изображение
        shutil.move(src_image, dst_image)
        # Перемещаем аннотацию, если она существует
        if os.path.exists(src_label):
            shutil.move(src_label, dst_label)
# Запускаем перенос файлов по каждому из трёх наборов
for split_name, files in dataset_splits.items():
    move_files(files, split_name)
# Выводим сообщение по завершению
print("Разделение данных завершено!")

```

Приложение В

Детектирование твёрдых бытовых отходов

Обзор датасета

Настоящий датасет разработан для автоматизированного обнаружения твердых бытовых отходов (ТБО), с использованием беспилотных летательных аппаратов (БПЛА) и алгоритма YOLOv8. Основная масса изображений датасета получена посредством видеосъемки санкционированной свалки строительного мусора, расположенной на урбанизированной территории города Алматы, что обусловлено высокой доступностью, экономической целесообразностью и удобством сбора данных. Для расширения объема и повышения разнообразия выборки дополнительно включены изображения несанкционированных свалок, расположенных в природных зонах Алматинской области. Съемка проводилась в несколько этапов в разные времена года, что позволяет охватить сезонные вариации внешнего вида объектов.

Структура и параметры датасета:

- **Общее количество изображений:** 1362
- **Технические характеристики изображений:**
 - ♦ Формат: JPEG (.jpg)
 - ♦ Разрешение: 4000×2250 пикселей (≈9 МП)
- **Источник данных:** видеозаписи, снятые БПЛА DJI MAVIC Mini 2 (модель FC7203) в пределах Алматинской городской агломерации. Съёмка велась с разной высоты, затем видеозаписи были нарезаны на кадры.
- **Формат аннотаций:** YOLOv8 (bounding box)
- **Количество классов:** 1 (dump)
- **Разметка:** выполнена вручную с использованием сервиса Roboflow
- **Предобработка данных:** на исходные изображения аугментация не применялась.
- **Перспективы дальнейшего развития**

В дальнейшем планируется расширение объема датасета за счет включения дополнительных изображений, что позволит повысить точность детекции и улучшить обобщающие возможности алгоритмов.

Приложение Г

Протокол проведения съёмок искусственно размещённого мусора

Место проведения съёмок: Съёмки проводились на двух близко расположенных локациях в районе улицы Сатпаева, 9А:

- **Локация 1** — участок с высокими деревьями, ветви которых с некоторых ракурсов частично перекрывают размещённый мусор.

- **Локация 2** — территория с невысокими елями и кустарниками, создающими разнообразный фон для съёмок.

Описание условий съёмки по датам:

11 марта 2025 года (12.00 – 13.00)

Погодные условия: солнечно, контрастные тени частично перекрывают мусор.

Особенности ландшафта: поверхность покрыта слоем снега.

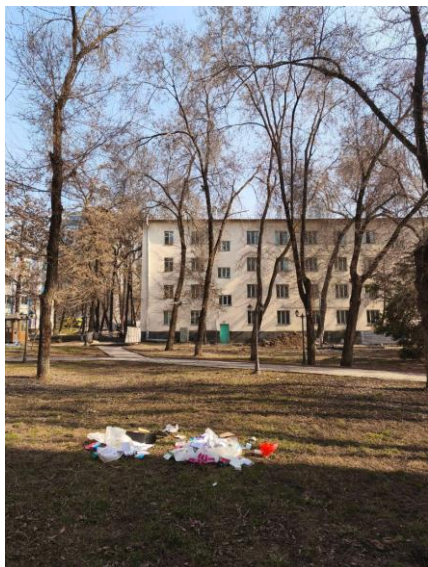
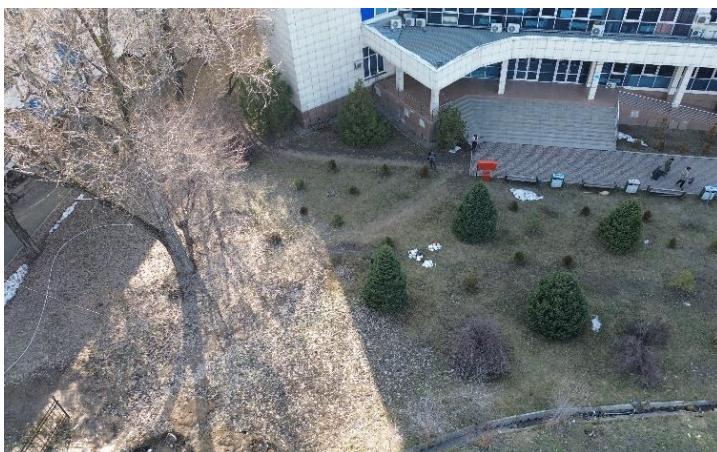
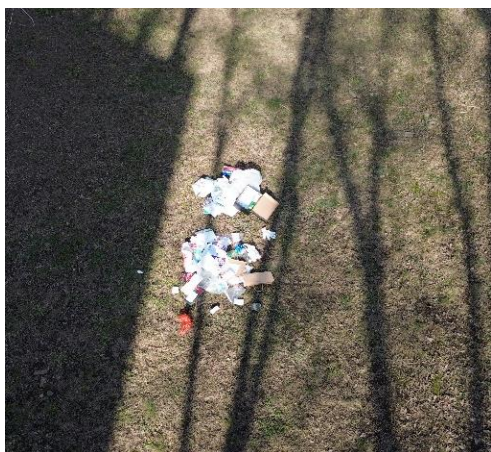
Проведено размещение объектов мусора на снежном покрове.



19 марта 2025 года (15.00 – 16.00)

Погодные условия: солнечно, но на второй локации мусор размещён в тени.

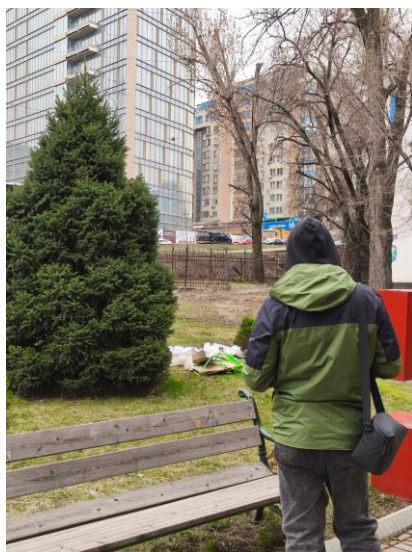
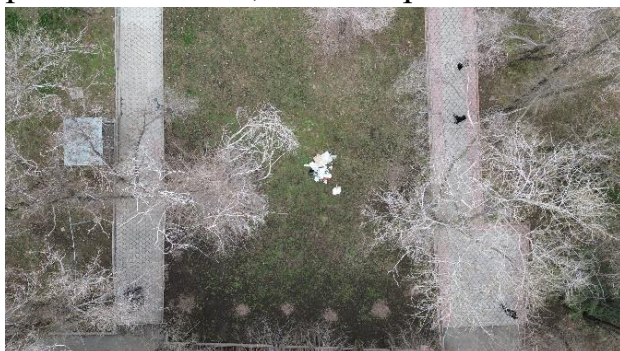
Особенности ландшафта: снег частично растаял, на участках остались отдельные кучи снега, которые визуальнo могут быть ошибочно восприняты как мусор.



27 марта 2025 года (14.00 – 15.00)

Погодные условия: пасмурно.

Особенности ландшафта: снег полностью растаял, появилась травянистая растительность, ветви деревьев остаются без листьев.



10 апреля 2025 года (14.00 – 15.00)

Погодные условия: солнечно.

Особенности ландшафта: сформировался плотный травяной покров, началось появление молодой листвы на деревьях.

