

DOI

УДК 631.34:004.896

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ И ИДЕНТИФИКАЦИЯ БОЛЕЗНЕЙ ПШЕНИЦЫ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ГЛУБОКОГО ОБУЧЕНИЯ И ПРИМЕНЕНИЕМ ДРОНОВ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ**С. Г. Мударисов, И. Р. Мифтахов**

Реферат. Исследование проводили с целью разработки методики раннего распознавания заболеваний пшеницы с использованием модифицированной версии модели YOLOv8n, направленной на достижение оптимального соотношения скорости и точности обнаружения в условиях реального времени. Для повышения точности регрессионного блока распознавания таких заболеваний, как бурая ржавчина, желтая ржавчина, плесень и септориоз, введена функция потери регрессии Glou bounding box. Предложена упрощенная сетевая структура YOLOv8n, адаптированная для использования в качестве магистральной сети с целью облегчения трансляции на мобильные терминалы. Применение методов предварительной подготовки, объединяющих смешанное и переносное обучение, способствует улучшению способности модели к обобщению. Для анализа использовали данные, собранные в ходе четырех полевых экспериментов на территории Уфимского и Кармаскалинского районов Республики Башкортостан. Всего в период полевого эксперимента было собрано более 10 000 изображений, из них для обучения моделей отобрано 1890 изображений листьев пшеницы. Обработка данных включала статистический анализ изображений, полученных в различных условиях освещения. Распознавание и оценку эффективности модели проводили с использованием показателей F1 и AP. F1-оценка при тестировании модели на изображениях, сделанных на фоне достаточного освещения и без укрытия листьями, находилась на уровне 54%, и AP-значение – 51,2%, при среднем значении IOU 50%. Точность идентификации болезни пшеницы на изображениях обучающего набора данных достигала 60%, тестового набора – 70%. Разработанная модель YOLOv8n для обнаружения заболеваний пшеницы в полевых условиях демонстрирует способность идентифицировать болезни в реальном времени с точностью 67,53%, что существенно превосходит величину этого показателя у других существующих моделей. В частности, точность YOLOv8n в 3 раза выше, чем у модели YOLOv4, что свидетельствует о значительном прогрессе в области автоматического обнаружения болезней растений.

Ключевые слова: зерновые культуры, болезни растений, искусственный интеллект, машинное обучение, беспилотный летательный аппарат, модель глубокого обучения.

Введение. Зерновые культуры занимают важное место в сельскохозяйственном производстве Российской Федерации и служат основным экспортным продуктом, позволяющим обеспечивать продовольственную безопасность. Основные факторы, ограничивающие их урожайность и снижающие качество продукции, наряду с вредителями и сорняками, – болезни растений.

Болезни растений представляют серьезную угрозу продовольственной безопасности по нескольким причинам. Во-первых, изменение климата создает новые условия для распространения и развития патогенов. Во-вторых, повышение мировой торговли продуктами питания способствует перемещению инфекций из одних регионов в другие, что увеличивает риск заражения новых территорий и культур. В совокупности эти факторы могут привести к снижению урожайности, ухудшению качества продукции и потере устойчивости сельскохозяйственных систем. По данным Продовольственной и сельскохозяйственной организации ООН (FAO), вредители и болезни растений приводят к потере от 20 до 40% мирового производства продовольствия [1].

Для эффективной борьбы с болезнями растений необходимо раннее обнаружение их симптомов. Традиционные методы определения, основанные на опыте специалистов, характеризуются низкой эффективностью,

подверженностью субъективному мнению и высокой трудоемкостью, что затрудняет практическое их применение [2, 3]. Появление машинного зрения, интернета вещей (IoT) и технологий дронов привело к значительному расширению использования инструментов искусственного интеллекта в сельском хозяйстве. Машинное зрение – технология, позволяющая компьютерам анализировать изображения без участия человека. Это автоматизирует обнаружение и классификацию объектов, особенно в сельском хозяйстве. Процесс включает захват изображений, их обработку с помощью соответствующих алгоритмов и идентификацию объектов [4].

В системах распознавания вредителей и болезней на основе дронов, машинное зрение играет важную роль, так как оно позволяет обрабатывать извлеченные изображения для определения таких характеристик объектов, как текстура, форма и цвет, на основе которых используются различные алгоритмы для классификации объектов.

В прошлом для классификации объектов требовалось ручное создание характеристик вредителей и болезней, а также классификаторов [5]. Это включало в себя применение таких подходов, как пороговая сегментация (разделение изображения на области), обнаружение краев с использованием методов типа Собеля, Робертса, Превитта, извлечение признаков,

включая масштабо-инвариантное преобразование (SIFT), гистограммы ориентированных градиентов (HOG), а также анализ формы, цвета и текстуры объектов.

Современные методы классификации включают использование таких алгоритмов машинного обучения, как машина опорных векторов (SVM), обратное распространение ошибки (BP), байесовский анализ и другие. Они позволяют автоматически классифицировать объекты на основе извлеченных характеристик, что улучшает точность и эффективность процесса обнаружения вредителей и болезней в сельском хозяйстве.

Удовлетворение требований к изображениям для обнаружения объектов в системах машинного зрения – сложная задача, требующая четкого различия между заболеваниями и окружающей средой [6, 7]. Существенные препятствия создают разнообразие типов заболеваний, их морфологические изменения на разных стадиях роста, а также возможность отображения схожих характеристик разными заболеваниями на разных стадиях. Эти сложности дополняются изменчивостью естественного света и сходством фона в реальных сценариях обнаружения, что может повлиять на надежность и эффективность традиционных алгоритмов распознавания с использованием машинного зрения.

В последние годы благодаря универсальности и гибкости в обработке сложных и огромных наборов данных во множестве научных дисциплин широкое распространение получили глубокие нейронные сети. Сверточные нейронные сети (СНС), рекуррентные нейронные сети (РНС), глубокие нейронные сети с прямым распространением (ГНС) и другие впервые привлекли внимание в 2012 году [8], когда с их использованием удалось сгенерировать содержательные выводы на основе аудиосигналов и визуальных изображений. Более поздние достижения, приведшие к созданию более надежных и эффективных нейронных сетей, сделали эти алгоритмы применимыми для решения спектра таких реальных задач, как наблюдение за транспортом [9], мониторинг массовых скоплений людей [10], анализ биомедицинских изображений [11], обнаружение болезней растений [12], наблюдения за реабилитацией пациентов [13]. Алгоритмы глубокого обучения применяют при изучении дикой природы для обнаружения поз и идентификации видов животных [14].

В частности, была продемонстрирована возможность классификации 14 болезней культурных растений с использованием общедоступного набора данных из 54306 изображений, включающего здоровые и больные растения, с точностью 99,35% [15], идентификации 8 различных видов мотыльков с использованием расширенного набора данных из 57600 изображений и максимальным показателем $F1=0,93$ [16]. Кроме того, после обучения сетей VGG-16 и ResNet на наборе данных

из 111467 изображений, были идентифицированы 20 видов млекопитающих с точностью 87,5% [17].

Цель исследований – разработка интегрированной системы распознавания болезней культурных растений для сельскохозяйственных БПЛА с использованием технологий глубокого обучения и сверточных нейронных сетей.

Для ее достижения решали следующие задачи:

- разработка глубокого обучающего фреймворка, включающего такие критические модули, как съемка изображений, передача данных и идентификация болезней, для их быстрой и точной классификации;

- исследование технологий глубокого обучения, включая модели, алгоритмы и метрики оценки, применяемых в системах распознавания болезней пшеницы для сельскохозяйственных БПЛА;

- тестирование системы на различных наборах данных для оценки ее способности к обнаружению заболеваний пшеницы и портируемости в реальных полевых условиях;

- определение ограничений, связанных с недостаточным набором данных, и разработка рекомендаций для дальнейших исследований, включая расширение и улучшение наборов данных для повышения стабильности и адаптивности алгоритмов.

Условия, материалы и методы. В нашем исследовании в качестве модели глубокого обучения для системы распознавания болезней сельскохозяйственных культур, а именно пшеницы, адаптированной для использования в сочетании с дронами, была выбрана сверточная нейронная сеть (СНС) CSPDarknet53 модель YOLOv8. СНС, разработанная с учетом пространственной структуры и локальной корреляции изображений, что повышает эффективность их обработки [18, 19, 20].

Система распознавания болезней, разработанная для сельскохозяйственных дронов, структурирована по трехуровневой архитектуре [21]. Она включает систему распознавания, систему передачи данных и систему обработки данных (рис. 1).

В состав системы распознавания входят беспилотная авиационная система (БАС), наземная станция управления и мобильный клиент. Первый уровень этой архитектуры посвящен удовлетворению динамических требований к распознаванию заболеваний сельскохозяйственных культур с использованием БАС.

Беспилотная авиационная система (БАС) отслеживает и запускает основное приложение, отвечающее за сбор данных о состоянии сельскохозяйственных культур и распознавание заболеваний, без участия человека. Это достигается благодаря автоматическим алгоритмам управления и предустановленным маршрутам полета, которые позволяют дрону выполнять свои задачи автономно.

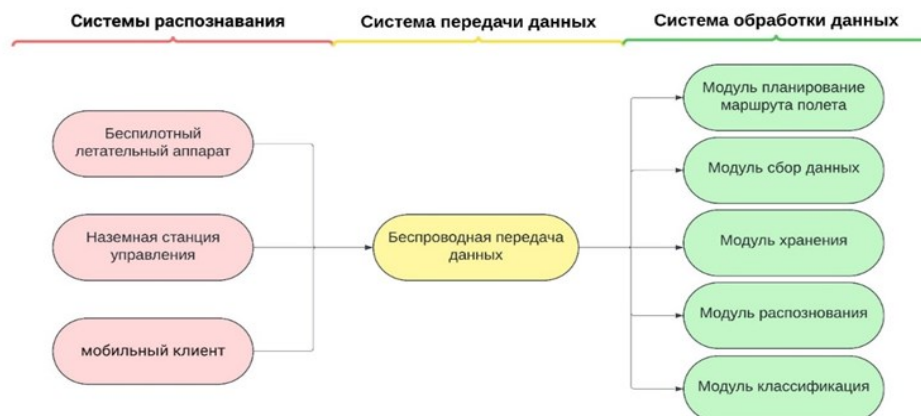


Рис. 1 – Структурная основа система распознавания болезней

Наземная станция осуществляет обработку данных, полученных дроном во время полета. Она принимает, сохраняет и анализирует эти данные, обеспечивая их интерпретацию и дальнейшее использование.

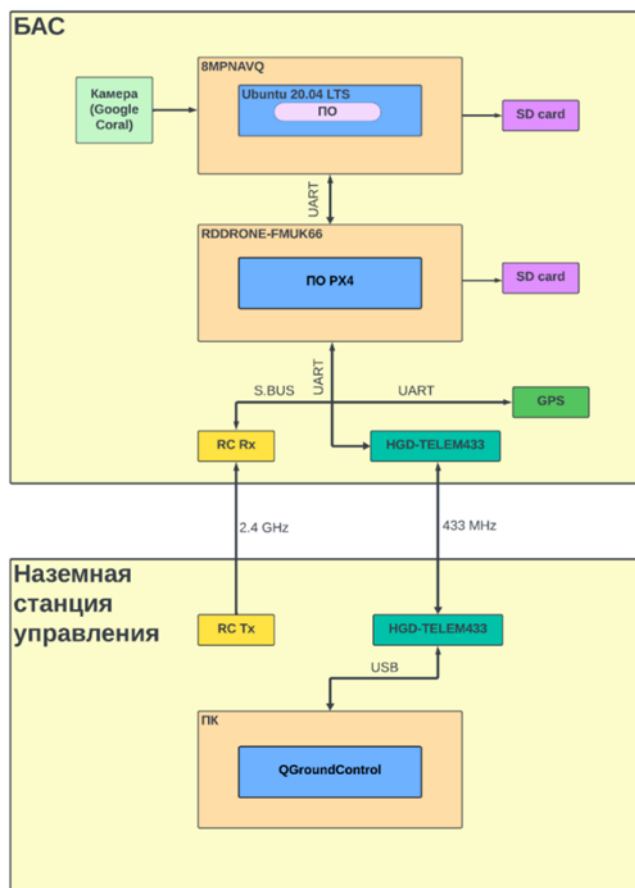


Рис. 2 – Архитектура системы распознавание заболеваний пшеницы

Архитектура системы распознавания болезней с использованием беспилотной авиационной системы (БАС) и наземной станции управления. БАС включает камеру (Google Coral), модуль 8MPNAVQ с Ubuntu 20.04 LTS, контроллер RDDRONE-FMUK66 с ПИО PX4, модули связи RC Rx и HGD-TELEM433. Наземная станция управления включает передатчик RC Tx, модуль HGD-TELEM433 и компьютер с ПО QGroundControl (рис. 2).

БАС была собран с использованием компонентов из набора NXP KIT-HGDRONEK66 (рис. 3). На ней установлены два компьютера: NXP RDDRONE-FMUK66 FMU (Flight Management Unit) и NavQPlus.

NXP RDDRONE-FMUK66 FMU (Flight Management Unit) – устанавливается на беспилотном летательном аппарате (БПЛА) и выполняет функции управления полетом. Он используется для запуска автопилота

PX4 – программного обеспечения управления полетом дрона, обеспечивающего мгновенный отклик на вводы и навигацию, а также для взаимодействия с различными периферийными устройствами, например, GPS и телеметрическим радио. По сравнению с NavQPlus, у него ограниченные ресурсы и поэтому он не так подходит для выполнения сложных операций.

NavQPlus – это AI/ML компьютер-помощник, который предназначен для запуска основного приложения на Python на беспилотных авиационных системах. Он обеспечивает вычислительные возможности для выполнения таких задач, как реализация алгоритмов компьютерного зрения, обработка данных и другие высокоуровневые операции. NavQPlus обычно оснащен специализированным аппаратным обеспечением с высокой производительностью и эффективностью работы алгоритмов машинного обучения и искусственного интеллекта.

Управление разработанной БАС осуществляется через протокол MAVLINK, который используется для передачи данных между БАС и его управляющим устройством, например, FMUK66 (Flight Management Unit). Протокол MAVLINK обеспечивает надежную передачу информации о состоянии БАС, команд управления, телеметрии и других

данных между различными устройствами в системе. Таким образом, через протокол MAVLINK управляющее устройство FMUK66 взаимодействует с наземной станцией для управления полетом и выполнения других функций, необходимых для работы БАС.

Наземная станция состоит из следующих компонентов:

ноутбук с установленной QGroundControl, который связывается с БАС через канал телеметрии HGD-TELEM433 (телеметрическое радио);

пульт дистанционного управления FS-i6S для удаленного управления БАС.

Для сбора входных данных в виде изображений или видео используется камера Google Coral, подключенная к NavQPlus (см. рис. 2). После получения данных о состоянии растений, компьютер использует обученную модель для их анализа и выполнения логического вывода относительно наличия или отсутствия заболевания [22, 23]. Кроме того, на компьютер передаются GPS-координаты местоположения пораженных растений. Они могут быть использованы для дальнейшего анализа или принятия решений, например, о проведении целевых обработок или мониторинга изменений состояния растений в течение времени.



Рис. 3 – Собранный БАС

Первым шагом при разработке прикладного программного обеспечения была настройка компьютера-компаньона NXP NavQPlus. При включении питания он автоматически загружается с поставленной SD-карты, на которой установлена обновленная версия операционной системы Linux - Ubuntu 20.04.5 LTS.

На компьютер-компаньон NavQPlus устанавливаются библиотеки PyTorch, NumPy и OpenCV. Перед началом работ через MAVLINK устанавливается связь с автопилотом PX4, затем БАС взлетает и через камеру (Google Coral) производится захват изображений растений с высоким разрешением, предоставляя входные данные для обработки и анализа (рис. 4). Далее производится запуск модели глубокого обучения для распознавания и классификации заболеваний растений.

Программа сама в автоматическом режиме будет производить обнаружение заболеваний растений на основе анализа изображений, включая классификацию и определение степени тяжести. А также программа производит получение географических координат местоположения для привязки результатов анализа к конкретным местам на поле. Затем обработанная информация поступает пользователю, а также эти данные сохраняются на SD-карте для последующего доступа и анализа.

Программное обеспечение (ПО) запускается с наземной станции, использование которой дает возможность контролировать операции (запускать, останавливать и др.). Управление БАС осуществляется вручную, все остальное работает автономно, без вмешательства пользователя.

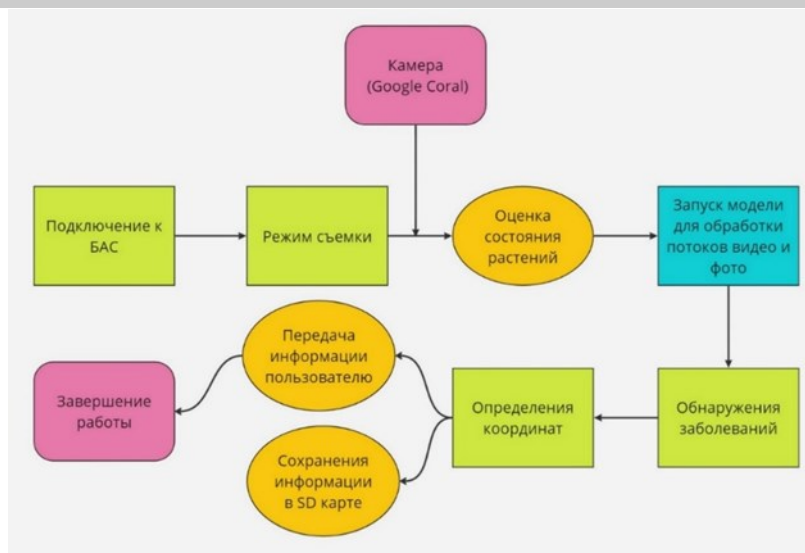


Рис. 4 – Архитектура программного обеспечения

Следует отметить, что основной сценарий программного кода main.py должен находиться на том же уровне, что и каталог YOLOv8n с обученной моделью, для того чтобы его выполнение проходило без ошибок.

Основные задачи в рамках вычислительно-го конвейера при анализе заболеваний растений – их обнаружение и отслеживание. Существующие алгоритмы обнаружения обычно разделяют на две категории: двух этапные, которые выделяют области интереса на изображениях и затем классифицируют их, например, Faster-RCNN, и одно этапные, такие как YOLO, SSD и DeepSort, которые используют методы регрессии для минимизации функций ошибок классификации и локализации. Последние работают быстрее и более эффективно, поскольку одновременно выполняют задачи обнаружения объектов и их отслеживания. Однако для эффективного обнаружения заболеваний растений в реальном времени такие модифицированные алгоритмы, как YOLO, SSD и DeepSort, требуют высокопроизводительных графических процессоров. При тестировании на встроенных платформах,

используемых в мобильных роботах, скорость обнаружения объектов значительно снижается. В основном это связано с производительностью графического процессора на встроенной платформе, которая ниже, чем у рабочей станции, что приводит к увеличению времени обработки.

Качественный набор данных – фундаментальное требование при создании модели глубокого обучения. Поэтому первый и решающий этап глубокого обучения начинается с получения изображений.

В нашем исследовании используются данные, собранные в ходе четырех полевых экспериментов в вегетационный период 2023 года на территории Уфимского и Кармаскалинского районов Республики Башкортостан (табл. 1). Сбор изображений осуществляли в течение 9 недель с использованием квадрокоптера Phantom 4 Pro с высококачественной RGB-камерой. Затем их вручную аннотировали и использовали в качестве обучающего набора. Кроме того, в обучающий набор были добавлены данные, собранные в открытых источниках интернет-ресурсов.

Таблица 1 – Общее количество собранных данных в период полевого эксперимента

Дата съемки	Число изображений, шт.	Размер файла
11.07.2023 г.	2158	4,3 Гб
26.07.2023 г.	2043	4,7 Гб
17.08.2023 г.	2314	4,4 Гб
30.08.2023 г.	2453	4,6 Гб
09.09.2023 г.	2012	4,1 Гб

Для разработки алгоритма распознавания была создана обширная база изображений, охватывающая разнообразие симптомов заболеваний. Их записывали в экспериментальных контролируемых и полевых условиях с использованием более чем 3 сортов и двух уровней удобрения (0 и 180 кг N/га), в различных фазах роста пшеницы и этапах заболевания листьев, в различных условиях освещения, с разными расстояниями между объектом и

объективом камеры (то есть пространственным разрешением). Изображения сохраняли в цветовом пространстве RGB в формате JPG и PNG. Для обеспечения достоверности информации о каждом пятне на листьях, используемом для обучения алгоритма по дальнейшей обработки.

Для классификации болезней доступны различные подходы [23]. После различных этапов предварительной обработки, например,

преобразования в другое цветовое пространство или установления порогового значения, из изображений извлекаются особенности [24], например, цвета, формы или текстуры [25], которые затем классифицируются с использованием таких инструментов машинного обучения, как машины опорных векторов (SVM) [26] или нейронные сети.

Данные, собранные по пшенице, позволили выделить следующие виды заболеваний (табл. 2, рис. 5):

бурая ржавчина (Brown rust) – возбудитель *Puccinia recondita* f. sp. *tritici*, проявляется в виде коричневых пятен с тысячами спор на листьях, стеблях и колосках пшеницы;

желтая ржавчина (Yellow Rust) – возбудитель *Puccinia striiformis* f. sp. *tritici*, проявляется в виде желтых полос на листьях пшеницы, которые со временем превращаются

в характерные ржавые пятна;

плесень пшеницы (Wheat Mold) – вызывается различными видами грибов, включая *Fusarium* spp., *Alternaria* spp. и *Aspergillus* spp., симптомы включают появление на листьях, стеблях или колосках характерного налета, который может быть белым, серым, черным или зеленым. Это поражение часто сопровождается гниением и разложением тканей растения;

септориоз (Septoria Leaf Blotch) – вызывается грибами рода *Septoria*, чаще всего видами *Septoria tritici* и *Septoria nodorum*, проявляется в виде характерных круглых или овальных бурых пятен на листьях пшеницы, которые обычно имеют серую или белую окантовку. При продолжительном заражении листья могут желтеть, высыхать и опадать.

Таблица 2 – Характеристики болезней листьев пшеницы

Заболевание	Патоген	Симптомы		
		форма патогена	диаметр	цвет
Бурая ржавчина	<i>Puccinia recondita</i> f. sp. <i>tritici</i>	различная	различный	проявляется в виде коричневых пятен
Желтая ржавчина	<i>Puccinia striiformis</i> f. sp. <i>tritici</i>	овальная	1...5 мм	желтые полосы, со временем превращаются в ржавые пятна
Плесень пшеницы	<i>Fusarium</i> spp., <i>Alternaria</i> spp. и <i>Aspergillus</i> spp.	круглая	2...5 мм	характерный налет, который может быть белым, серым, черным или зеленым
Септориоз	<i>Septoria tritici</i> и <i>Septoria nodorum</i>	овальная	2...10 мм	проявляется в виде бурых пятен



Рис. 5 – Изображения листьев пшеницы, сделанные с помощью БАС, показывают типичные пятна на листьях, вызванные 4 различными патогенами

Все изображения по каждому заболеванию были распределены между тренировочным (train), валидационным (val) и тестовым (test) наборами в соотношении 70%, 20% и 10% соответственно (табл. 3).

Такое распределение довольно часто используют при глубоком обучении. Наименьшее количество фотографий было в классе «бурая ржавчина», наибольшее – «септориоз». Неравномерное распределение классов может повлиять на обучение модели, особенно если она будет лучше распознавать классы

с большим количеством данных. Кроме того, необходимо, чтобы распределение классов в валидационном и тестовом наборах было сопоставимо с тренировочным.

Для начала обучения необходимо провести предварительную обработку полученных данных, которая относится к действиям, выполняемым на фундаментальном уровне [27, 28].

Обучение моделей происходит быстрее при меньшем размере изображений, выраженном в пикселях, поскольку это означает меньшее количество пикселей для обработки.

Таблица 3 – Соотношения количество собранных данных для обучения

Класс заболеваний	Количество фото	train	val	test
Бурая ржавчина	161	113	32	16
Плесень	517	362	103	52
Желтая ржавчина	429	300	86	43
Септориоз	753	527	151	75
Процентное соотношения		70%	20%	10%

Кроме того, исходные изображения могут иметь разные размеры, а многие архитектуры моделей глубокого обучения требуют, чтобы они были одинаковыми [29]. Кроме того, при разнице между обучающим изображением и изображением распознавания модель обучения YOLO показывает плохую производительность [30]. Так как первоначальные размеры изображений были разными в нашем обучающем наборе захваченные исходные изображения масштабировали до размеров 640x640 пикселей для обеспечения согласованности данных и оптимальной работы модели. Такие размеры считаются оптимальным для многих версий модели YOLO. Этот подход позволяет сбалансировать скорость и точность обучения, так как меньшие изображения обрабатываются быстрее, но могут содержать меньше деталей. Изменение размера изображений до 640x640 пикселей обеспечивает оптимальный компромисс между скоростью обработки и сохранением достаточного количества деталей для точного распознавания заболеваний. Кроме того, также размер изображения должен быть кратен 32, потому что в архитектуре YOLO используются сверточные слои с уменьшением размерности (downsampling). В каждом таком слое размер изображения уменьшается вдвое. Начиная с 640, после нескольких уменьшений мы по-прежнему получаем целочисленные размеры, что важно для корректной работы сверточной сети. Если размер не будет кратен 32, может возникнуть необходимость в нецелочисленном количестве пикселей, что неприемлемо для сверточной архитектуры. При изменении

размера изображения его пиксельные значения уменьшаются, и нежелательная область интереса отбрасывается. Такой метод предварительной обработки выполняется с использованием скрипта на Python и библиотеки OpenCv перед обучением модели обнаружения и отслеживания объектов (YOLO).

Аннотирование изображений – это процесс преобразования неструктурированных данных в структурированную форму, понятную для глубокого обучения. Аннотированный (размеченный) набор данных – наиболее важная часть машинного обучения (ML). Инженеры машинного обучения в исследованиях используют такие инструменты аннотации, как labelling, roboflow, yolo_mark и многие другие.

Для аннотации данных YOLOv8 мы использовали инструмент CVAT, который бесплатно доступен в репозитории компании CVAT. В то время как roboflow, являющийся платно платформой компьютерного зрения, позволяет создавать модели компьютерного зрения и аннотации изображений в сочетании с аугментацией данных.

ПО CVAT было развернуто на локальном Docker контейнере, что обеспечивает удобство использования, скорость и эффективность аннотирования данных, а также сохранность конфиденциальности данных, поскольку всё происходит на вашем собственном компьютере или сервере.

Аннотирование с использованием программного комплекса CVAT представляет собой процесс разметки изображений, на которых видны симптомы различных заболеваний (рис. 6).

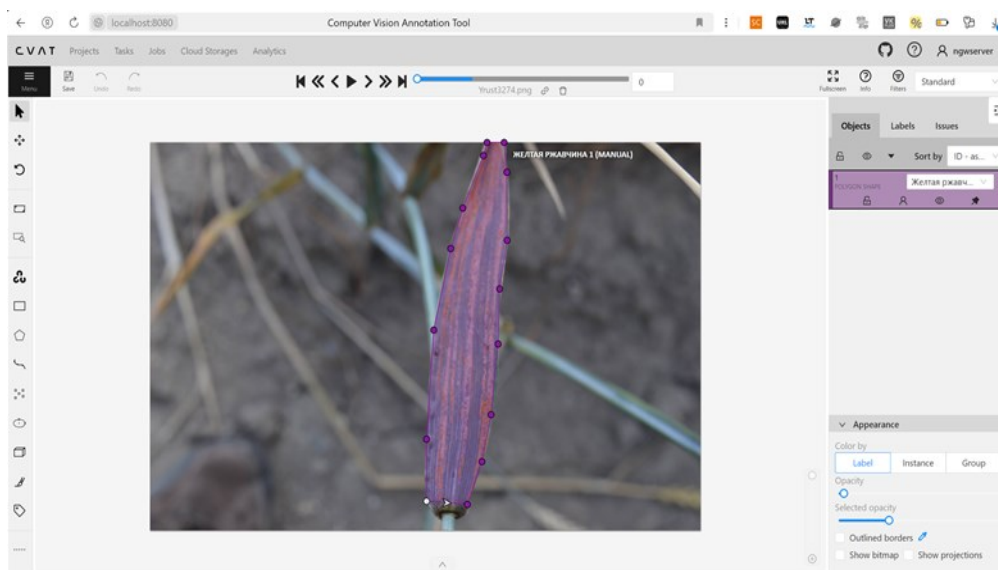


Рис. 6 – Процесс нанесение разметки

В результате создается набор данных с размеченными симптомами заболеваний, который можно использовать для машинного обучения их автоматического обнаружения и классификации на изображениях.

В нашем исследовании использовался Python версии 3.9.18 с библиотекой PyTorch версии 2.0.1, которая поддерживает CUDA версии 11.7. PyTorch был оптимизирован для работы с GPU, используя CUDA, что позволяет значительно ускорить вычисления благодаря параллельной обработке данных на GPU. В качестве оборудования использовали видеокарту NVIDIA GeForce RTX 3070 с 8 ГБ памяти, что позволяло проводить сложные вычисления, связанные с тренировкой и использованием моделей машинного обучения, эффективно и быстро. Обучение модели проводили на протяжении 700 итераций с размером пакета (batch size) равным 2. Для оптимизации использовали оптимизатор Adam. Начальная скорость обучения была установлена на уровне 0,001 (1e-3), максимальная была задана как 0,00001 (1e-5). Коэффициент импульса (momentum) составлял 0,937, снижение веса (weight decay) – 0, что означает отсутствие регуляризации по весам.

Все варианты YOLO демонстрируют высокую эффективность в классификации заболеваний пшеницы. Однако, согласно опубликованным исследованиям, YOLOv8n выделяется среди прочих версий благодаря повышенной точности и эффективности решения такой задачи. Этот факт делает YOLOv8n более предпочтительной для реализации в проектах, связанных с диагностикой заболеваний пшеницы, что подтверждают результаты исследований других авторов [31]. В качестве показателей оценки эффективности модели использовали такие метрики, как точность классификации, средняя точность по всем классам (mAP), потери в процессе обучения и полнота [32].

Существуют два типа потерь: один представляет собой потери обучения детекции, другой – потери валидации детекции каждого объекта. При проведении исследований в течение каждой эпохи оценивали потери обучения и после каждой эпохи определяли потери тестирования.

Способность модели правильно классифицировать объекты определенного класса или ее точность рассчитывается по формуле:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (1)$$

где TP (True Positives) — количество истинно положительных результатов, это число случаев, когда модель корректно определила наличие класса заболеваний;

FP (False Positives) — количество ложно положительных результатов, это число случаев, когда модель ошибочно определила наличие класса заболеваний.

Способность модели обнаруживать все

действительно положительные объекты целевого класса оценивается метрикой Recall (полнота). Она характеризует долю правильно обнаруженных объектов целевого класса относительно их общего количества в тестовом наборе данных. Полнота рассчитывается по формуле:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (2)$$

FN (False Negatives) — количество ложно отрицательных результатов, когда модель не смогла обнаружить наличие класса заболеваний.

Метрика mean Average Precision (mAP) представляет собой среднее значение точности (Precision) для каждого класса или категории объектов. Он характеризует компромисс между точностью и полнотой и рассчитывается по следующей формуле:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (3)$$

где N – количество классов, AP – среднее значение метрик по всем классам.

Функция потерь модели YOLO рассчитывается по формуле:

$$\text{LOSS} = \text{LOSS}_{cls} + \text{LOSS}_{conf} + \text{LOSS}_{reg}, \quad (4)$$

где LOSS – функция потерь модели YOLO, индекс cls – потери классификации, conf – потери уверенности, reg – потери регрессии (пикселях).

В свою очередь потери классификации (cls), уверенности (conf) и регрессии (reg), которые используются для оценки ошибки модели рассчитываются по следующим формулам:

$$\text{LOSS}_{cls} = \sum_{i=0}^{k^2} li \sum_{c=1}^c E(\hat{p}_i(c), p_i(c)) \quad , \quad (5)$$

$$\text{LOSS}_{cls} = \sum_{i=0}^{k^2} \sum_{j=0}^m l_{i,j} E(\hat{c}_i, C_i) - \quad , \quad (6)$$

$$- \lambda_{no\ obj} \sum_{i=0}^{k^2} \sum_{j=0}^m l_{i,j} E(1, I_{i,j}) E(\hat{c}_i, C_i) \\ \text{LOSS}_{reg} = 1 - \text{IOU}(A^p, A^\delta) + \frac{A^c - A^p - A^\delta + 1}{A^c} \quad , \quad (7)$$

где k^2 — это размер сетки; Если изображение разделено на сетку $k \times k$, то общее количество ячеек сетки будет k^2 ;

li – индикаторная функция, которая равна 1, если в ячейке i сетки присутствует объект, и 0 в противном случае;

$\sum_{c=1}^c E$ – сумма ошибок для каждого класса c в ячейке сетки i , где C — это количество классов;

$(\hat{p}_i(c), p_i(c))$ – идентифицированная вероятность и истинная вероятность того, что объект в ячейке i принадлежит классу c ;

m – предположительно количество

идентифицированных рамок в каждой ячейке сетки;

$I_{i,j}$ – индикаторная функция, которая равна 1, если предсказанная рамка j в ячейке сетки i отвечает за обнаружение объекта;

(\hat{C}_i, C_i) – идентифицированная и истинная метки классов для рамки j в ячейке i ;

λ_{noobj} – весовой коэффициент для потерь, связанных с отсутствием объекта;

$I_{i,j}$ – индикаторная функция, равная 1, если идентифицированная рамка j в ячейке i не отвечает за обнаружение объекта;

$IOU(A^p, A^\delta)$ – пересечение по объединению (Intersection over Union) между идентифицированными рамками A^p и истинными рамками A^δ ;

A^C – площадь объединения всех истинных и предсказанных рамок;

A^p и A^δ – площадь идентифицированных и истинных рамок соответственно;

I – площадь пересечения между A^p и A^δ .

Чем ниже потери модели, тем выше ее производительность.

Так же при проведении исследований определяли балл уверенности, который обычно рассчитывается в процессе прямого распространения (forward propagation) модели. После того, как модель обработала входные данные (например, изображение с растением), она генерирует прогнозы на основе изученных признаков и весов. Для каждого потенциального обнаружения модель выдаёт два основных значения:

классификация – к какому классу принадлежит обнаруженный объект;

балл уверенности – вероятность того, что обнаруженный объект действительно принадлежит указанному классу.

Балл уверенности (C) для каждого прогноза может быть рассчитан по формуле (8), где P означает вероятность принадлежности обнаружения к определенному классу:

$$C = \frac{\text{объект принадлежит классу}}{\text{объект обнаружен}} \quad (8)$$

Эта вероятность обычно вычисляется с использованием такой функции активации, как сигмоид, которая преобразует выходные данные модели в вероятность между 0 и 1.

В контексте обнаружения объектов, балл уверенности также может быть скомбинирован с точностью ограничивающей рамки, что дает более общий балл уверенности для каждого обнаруженного объекта.

В данном исследовании, модель YOLOv8n обучается, начиная с инициализации параметров модели значениями из обученного файла весов оригинальной модели YOLOv8n. Это означает, что вместо начала обучения с случайно инициализированных весов, исследование использует веса, уже предварительно обученные на другом наборе данных или задаче, что помогает ускорить процесс обучения и потенциально улучшить производительность модели. Поскольку для сравнительного эксперимента требуется обучение различных структур сетей и определение оптимального количества итераций для достижения наилучшей производительности обнаружения, в ходе исследования динамически отслеживался процесс обучения и сохранялся файл весов сети на каждой эпохе. Это позволило выбрать лучшую обучающую модель и предотвратить переобучение.

Для динамического наблюдения за статусом обучения и производительностью модели на различных итерациях использовали инструмент Wandb (Weights & Biases).

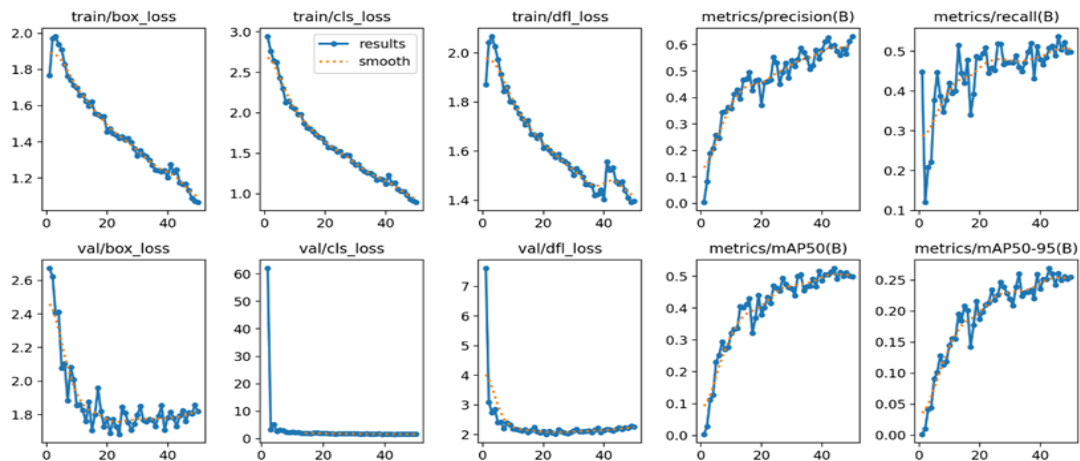


Рис. 7 – Визуализация оценки производительности модели YOLOv8n: train/box_loss – потери от регрессии ограничивающих рамок во время обучения (определяет насколько хорошо модель способна точно прогнозировать координаты ограничивающих рамок вокруг объектов); train/cls_loss – измеряет, насколько хорошо модель идентифицирует правильные классы объектов на изображении; train/dfl_loss – потери от уровней пирамиды признаков с деформируемыми свертками (DFL), позволяющие модели лучше адаптироваться к деформациям объектов, учитывая несбалансированные наборы данных; val/box_loss, val/cls_loss, val/dfl_loss – оценка на валидационном наборе данных

Анализ процесса обучения и валидации используемой модели машинного обучения (рис. 7) дает возможность сделать следующие заключения:

модель демонстрирует улучшение способности к локализации объектов, что подтверждается уменьшением потерь ограничивающей рамки (Box Loss) с начальных значений около 2.0 на этапе обучения и 2.6 на этапе валидации до примерно 1.2 и 1.8 соответственно;

отмечено значительное улучшение точности классификации объектов благодаря снижению потерь классификации (Cls Loss) с начальных 3.0 до около 0.5 для обучающей выборки и с более чем 50 до значений, близких к нулю, для валидационной выборки;

потери, связанные с Dfl, также показывают тенденцию к улучшению, снижаясь с примерно 2.0 до 1.5 на этапе обучения и с более чем 7 до около 1 на этапе валидации;

метрики точности (Precision) и полноты (Recall) продемонстрировали увеличение со значений около нуля до стабилизации примерно на уровне 0.6 и 0.5 соответственно, что указывает на баланс между способностью модели корректно классифицировать положительные примеры и её способностью обнаруживать все релевантные случаи;

показатели средней точности идентификации на различных уровнях порога IoU также улучшились, с mAP50 увеличившимся с начальных значений до более чем 0.5 и mAP50-95 с нуля до приблизительно 0.25, что свидетельствует об улучшении общей точности обнаружения и классификации моделью.

В исследовании, проведенном Сафари Джалал Бехзадом и соавторами, было продемонстрировано, что при детекции таких болезней, как желтая и бурая ржавчина, модель YOLOv4 tiny обеспечила среднюю точность (mAP) на уровне 19% [33]. В нашем исследовании использованием модели YOLOv8n позволило достичь средней точности (mAP) равной 65%. Эти результаты наглядно демонстрируют превосходство модели YOLOv8n, по сравнению с моделью YOLOv4 tiny.

Производительность предиктора отражает, насколько эффективно алгоритм машинного обучения справляется с задачей прогнозирования или предсказания результатов. В контексте YOLOv8n для классификации, обнаружения, сегментации и отслеживания болезней, эта эффективность измеряется с использованием функции потерь (4). Она используется для оценки корреляции между предоставленными данными и ожидаемыми результатами. Результаты работы модели YOLOv8n (рис. 8) представляют собой визуальную проверку способности модели обнаруживать и классифицировать такие заболевания пшеницы, как бурая ржавчина, желтая ржавчина, септориоз и плесень. Результаты тестирования, проведенного для проверки работоспособности разработанной модели на примере септориоза, показали, что она правильно и эффективно идентифицирует заболевание с более высоким уровнем уверенности, по сравнению с другими моделями (рис. 8). Пороговое значение было установлено на уровне 0,3, то есть при балле уверенности 30% и более модель классифицирует объект в соответствующий класс.



Рис. 8 – Результаты работы модели YOLOv8n

Высокий балл уверенности в идентификации заболевания септориоза 0.9 означает, что модель с высокой вероятностью считает диагностирование правильным (см. рис. 8). Такие результаты тестирования указывают на то, что разработанная модель эффективно выявляет симптомы септориоза на изображениях растений, что важно, например, в системах автоматической диагностики в агросекторе.

Для повышения эффективности алгоритмических операций при одновременной минимизации массовых характеристик полезной нагрузки беспилотных авиационных систем были выбраны высокопроизводительные, но легковесные аппаратные компоненты. В контексте рассматриваемой задачи, для этих целей лучше всего подходил микропроцессорный модуль NXP 8MPNAVQ компании NXP Semiconductors, который характеризуется оптимальным соотношением вычислительной мощности и энергопотребления.

Результаты анализа аппаратных характеристик демонстрируют, что коммерчески доступные нейронные модули и аппаратные

ускорители типа NXP 8MPNAVQ (NavQPlus AI/ML компьютер-помощник) обладают преимуществами в виде упрощенной интеграции в систему БАС и сниженного энергопотребления. Однако их применение в приложениях реального времени ограничивает низкая частота кадров. Поэтому была выбрана платформа NXP 8MPNAVQ, которая может работать с Ubuntu Linux и поддерживает ROS2. Процессор NXP 8MPNAVQ, функционирующий на четырех ядрах Arm Cortex-A53 с максимальной тактовой частотой до 2,0 ГГц и оснащенный кэш-памятью ECC 512 КБ, обеспечивает вычислительную мощность до 472 GFlops. При этом его номинальная масса составляет 100 г, а потребление энергии варьируется в пределах 5...10 ватт, что делает его подходящим для беспилотных систем, где каждый грамм и каждый ватт имеют значение.

Способность устройства обеспечить скорость обнаружения в 8 кадров в секунду, хотя и не идеальна, оказывается приемлемой в контексте заданных ограничений.

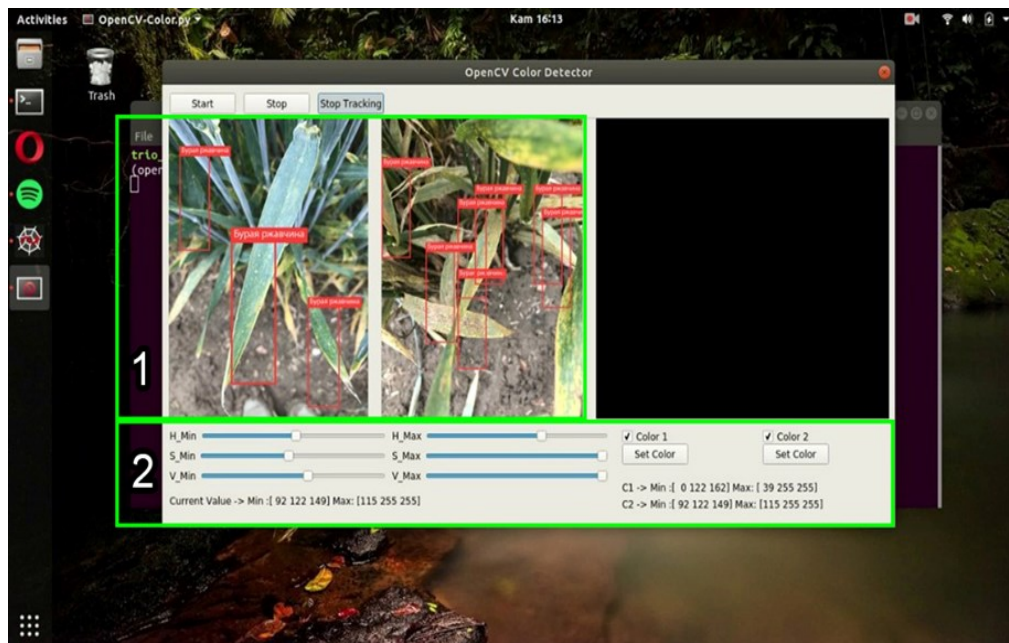


Рис. 9 – Демонстрация работы компьютерной программы

В процессе работы компьютерной программы с левой стороны отображается видео с камеры, с правой – статическое изображение с ограничительными рамками вокруг обнаруженных объектов и названиями обнаруженных заболеваний (рис. 9). Для обнаружения цветов на изображениях в программе используется библиотека OpenCV с моделью цвета HSV. В графическом пользовательском интерфейсе есть ползунки для установки минимальных и максимальных значений оттенка – «H», насыщенности – «S» и яркости – «V».

В целом, несмотря на обширные исследования по обнаружению желтой и бурой ржавчины с использованием методов глубокого обучения, до сих пор они не полностью

учитывали сложности полевых условий и различные углы, под которыми БАС может наблюдать симптомы этих заболеваний [33, 34, 35]. Кроме того, некоторые исследования проводили с использованием классического машинного обучения. Сведений о работах по обнаружению септориоза с применением БАС и глубокого обучения в доступной авторам зарубежной и отечественной литературе не найдено. Результаты проведенных исследований устраняют эти пробелы и подчеркивают значительный потенциал модели YOLOv8n в идентификации септориоза, плесени, желтой и бурой ржавчины.

Проведенные исследования показали несколько явных преимуществ модели

YOLOv8n:

для работы модели необходимо меньше вычислительных ресурсов, что делает ее более подходящей для использования в таких системах с ограниченной вычислительной мощностью, как БАС или периферийные устройства;

архитектура модели достаточно простая и мощная для обеспечения высокой точности обнаружения объектов;

использование ориентированных ограничивающих рамок увеличивает точность идентификации и локализации симптомов таких заболеваний, как желтая и бурая ржавчина, минимизируя при этом влияние избыточной фоновой информации, что особенно актуально при анализе данных, полученных с беспилотных летательных аппаратов.

Для дальнейшего углубления аналитического потенциала данного подхода предлагаются разработка методик для количественного описания степени поражения, включая алгоритмы для измерения площади пораженных участков и оценки интенсивности заболевания.

При этом величина коэффициента полноты модели YOLOv8n для обнаружения

септориоза, плесени, желтой и бурой ржавчины составляет 60,0%, что указывает на необходимость его повышения. Такие факторы, как воздействие ветра, размытие из-за винтов БПЛА и близкого расположения симптомов заболевания, приводят к пропускам при их детекции. Для решения этих проблем в дальнейших экспериментах запланировано изучение возможности использования адверсативных генеративных сетей.

Выводы. Успешная интеграция аппаратных средств с вычислительными алгоритмами позволило эффективно детектировать заболевания пшеницы прямо в поле, демонстрируя практическую ценность разработанной системы.

Предложенная модель основана на алгоритме YOLOv8 (You Only Look Once version 8) и позволяет обнаруживать болезни пшеницы в реальном времени не зависимо от угла обзора БАС.

Алгоритм модели глубокого обучения способен распознавать целевые объекты с высокой точностью, что позволяет применять его при детекции различных видов заболеваний культурных растений.

Литература

1. FAO. «The Impact of Pests and Diseases on Plant Productivity». URL: <https://www.fao.org/3/i7829ru/17829RU.pdf> (дата обращения: 12.03.2024).
2. Lee S., Arora A. S., Yun C. M. Detecting strawberry diseases and pest infections in the very early stage with an ensemble deep-learning model // *Frontiers in Plant Science*. 2022. Vol. 13. URL: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.991134/full> (дата обращения: 17.06.2024).
3. Kim J., Choi J., Lee D. Unmanned Aerial Vehicles in Agriculture: A Review of Perspective of Platform, Control, and Applications // *IEEE Access*. 2019. Vol. 7. pp. 105100-105115. URL: https://link.springer.com/chapter/10.1007/978-3-319-25479-1_29 (дата обращения: 17.06.2024).
4. Gokool S., Rampadarath H., and Bissessur P. Crop monitoring in smallholder farms using unmanned aerial vehicles to facilitate precision agriculture practices: a scoping review and bibliometric analysis // *Sustainability*. 2023. Vol. 15. №. 4. P. 3557.
5. Object detection based on global-local saliency constraint in aerial images / C. Li, B. Luo, H. Hong, et al. // *Remote Sensing*. 2018. Vol. 12 (9). URL: <https://www.mdpi.com/2072-4292/12/9/1435> (дата обращения: 12.03.2024).
6. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G. E., et al. // *Advances in Neural Information Processing Systems*. 2012. Vol. 25. P. 1097–1105. URL: <https://dl.acm.org/doi/abs/10.1145/3065386> (дата обращения: 12.03.2024).
7. Zhang S., Li H., Wang J. A real-time and lightweight traffic sign detection method based on ghost-YOLO // *Multimedia Tools and Applications*. 2023. Vol. 82. No. 17. P. 26063-26087.
8. DeepCount: crowd counting with Wi-Fi using deep learning / Y. Zhao, S. Liu, F. Xue, et al. // *Journal of Communications and Information Networks*. 2019. Vol. 4(3). P. 38–52.
9. Deep learning in biomedical image analysis / M. Kim, C. Yan, D. Yang, et al. // *New York: ACM Press*, 2020. URL: <https://dl.acm.org/doi/abs/10.1145/3065386> (дата обращения: 17.06.2024).
10. Vidhyalatha T., Sreeram Y., Purushotham E. Animal Intrusion Detection using Deep Learning and Transfer Learning Approaches // *International Journal of Human Computations & Intelligence*. 2022. Vol. 1. No. 4. P. 51–60. URL: <https://milestoneresearch.in/JOURNALS/index.php/IJHCI/article/view/51> (дата обращения: 17.06.2024).
11. Palani P., Kumar S., and Singh R. Real-time joint angle estimation using mediapipe framework and inertial sensors // *22nd International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, 2022. P. 128-133.
12. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning / M.S. Norouzzadeh, A. Nguyen, M. Kosmala, et al. // *Proc Natl Acad Sci U S A*. 2018. Vol. 115. No. 25. E5716-E5725. URL: <https://www.pnas.org/doi/full/10.1073/pnas.1719367115> (дата обращения: 17.06.2024). doi: 10.1073/pnas.1719367115.
13. Mohanty S.P., Hughes D.P., Salathé M. Using deep learning for image-based plant disease detection // *Frontiers in Plant Science*. 2016. Vol. 7. P. 1419. URL: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2016.01419/full> (дата обращения: 12.03.2024).
14. An automated light trap to monitor moths (Lepidoptera) using computer vision-based tracking and deep learning / K. Bjerge, J.B. Nielsen, M.V. Sepstrup, et al. // *Sensors*. 2022. Vol. 21(2). P. 343. URL: <https://www.mdpi.com/1424-8220/21/2/343> (дата обращения: 12.03.2024).
15. Insights and approaches using deep learning to classify wildlife / Z. Miao, K.M. Gaynor, J. Wang, et al. // *Scientific Reports*. 2019. Vol. 9(1). P. 8137. URL: <https://www.nature.com/articles/s41598-019-44565-w> (дата обращения: 12.03.2024).
16. Confronting deep-learning and biodiversity challenges for automatic video-monitoring of marine ecosystems

tems / S. Villon, C. Iovan, M. Mangeas, et al. // *Sensors*, 2022. Vol. 22(2). P. 497. URL: <https://www.mdpi.com/1424-8220/22/2/497> (дата обращения: 12.03.2024).

17. Vehicle detection and tracking in adverse weather using a deep learning framework / M. Hassaballah, M.A. Kenk, K. Muhammad, et al. // *IEEE Transactions on Intelligent Transportation Systems*. 2020. Vol. 22(7). P. 4230–4242.

18. GBSG-YOLOv8n: A Model for Enhanced Personal Protective Equipment Detection in Industrial Environments / C. Shi, D. Zhu, J. Shen, et al. // *Electronics*. 2023. Vol. 12. 4628. URL: <https://www.mdpi.com/2079-9292/12/22/4628> (дата обращения: 12.03.2024).

19. Agarwal M., Gupta S., Biswas K. K. A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant // *Sustainable Computing: Informatics and Systems*. 2021. Vol. 30. P. 100473. Режим доступа: <https://www.sci-hub.ru/10.1016/j.suscom.2020.100473>, (дата обращения: 17.06.2024).

20. Чичкова Е. Ф., Кочин Д. А., Рогачев С. А. Информационно-аналитический сервис и технологии мониторинга потенциально опасных зон по данным спутникового многоспектрального зондирования // *Космические аппараты и технологии*. 2023. Т. 7. №. 2. С. 132-141.

21. A mobile-based system for maize plant leaf disease detection and classification using deep learning / M. Abdullah, S. A. Shah, K. N. Saddozai, et al. // *Frontiers in Plant Science*. 2023. Vol. 14. No. 1079366. URL: <https://www.frontiersin.org/articles/10.3389/fpls.2023.1079366/full> (дата обращения: 17.06.2024).

22. Astani M., Hasheminejad M., Vaghefi M. A diverse ensemble classifier for tomato disease recognition // *Computers and Electronics in Agriculture*. 2022. Vol. 198. P. 107054. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0168169922003714?via%3Dihub> (дата обращения: 12.03.2024).

23. Austria Y. C., Mirabueno M. C. A., Lopez D. J. D. EZM-AI: A YOLOv5 machine vision inference approach of the philippine corn leaf diseases detection system // 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (ICAJET). IEEE, 2022. – P. 1-6. URL: <https://ieeexplore.ieee.org/document/9634287> (дата обращения: 17.06.2024).

24. A review of advanced machine learning methods for the detection of biotic stress in precision crop protection / J. Behmann, A. K. Mahlein, T. Rumpf, et al. // *Precis. Agric.* 2015. Vol. 16. P. 239–260.

25. Zhang L., Li Y., Xu X. Grape leaf disease detection using a deep learning-based approach // *Computers and Electronics in Agriculture*. 2022. Vol. 192. P. 106710. URL: <https://doi.org/10.1016/j.compag.2022.106710> (дата обращения: 12.03.2024).

26. Badiger M. et al. Leaf and skin disease detection using image processing // *Global Transitions Proceedings*. 2022. Vol. 3. No. 1. P. 272-278.

27. Zhang L., Li Y., Xu X. Artificial Driving-based EfficientNet for Automatic Plant Leaf Disease Classification // *Multimedia Tools and Applications*. 2022. Vol. 81. No. 17. P. 24039-24056. URL: <https://link.springer.com/article/10.1007/s11042-022-12345-6> (дата обращения: 12.03.2024).

28. Sarki A. Advancements in Drone Technology for Precision Agriculture // *Agricultural Technology Review*. 2021. Vol. 25. P. 65-78.

29. Sharma R. Integration of Unmanned Aerial Vehicles in Plant Disease Management // *International Journal of Plant Pathology*. 2020. Vol. 30. P. 145-158.

30. A Study of Ship Resistance Characteristics for Ice Strengthened Vessel by Broken Ice Channel Width and Size of Broken Ice Pieces / S.Y. Jeong, J. Jang, C.H. Kim, et al. // *J. of the Society of Naval Architects of Korea*. 2018. Vol. 55 (1). P. 22-27.

31. Na M. A., Li J., Zhang P. Improved YOLOv8-based automated detection of wheat leaf diseases // *INMATEH-Agricultural Engineering*. 2023. Vol. 71. No. 3. URL: <https://inmateh.eu/volumes/volume-71--no-3--2023/improved-yolov8-based-automated-detection-of-wheat-leaf-diseases/> (дата обращения: 17.06.2024).

32. Gai R., Chen N., Yuan H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model // *Neural Computing and Applications*. 2023. Vol. 35. No. 19. P. 13895-13906.

33. Safarjalal B., Alborzi Y., Najafi E. Automated wheat disease detection using a ROS-based autonomous guided UAV // *arXiv preprint arXiv: 2206.15042*. 2022. URL: <https://arxiv.org/abs/2206.15042> (дата обращения: 12.03.2024).

34. Bohnenkamp D., Behmann J., Mahlein A. K. In-field detection of yellow rust in wheat on the ground canopy and UAV scale // *Remote Sensing*. 2019. Vol. 11. No. 21. P. 2495. URL: <https://www.mdpi.com/2072-4292/11/21/2495> (дата обращения: 17.03.2024).

35. Su J., Chen D., Li W. Spatio-temporal monitoring of wheat yellow rust using UAV multispectral imagery // *Computers and Electronics in Agriculture*. 2019. Vol. 167. P. 105035. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0168169918318520?via%3Dihub> (дата обращения: 17.06.2024).

Сведения об авторах:

Мударисов Салават Гумерович – доктор технических наук, заведующий кафедры мехатронных систем и машин аграрного производства, e-mail: salavam@gmail.com

Мифтахов Ильнур Ринатович – младший научный сотрудник, e-mail: info323@bk.ru

Башкирский государственный аграрный университет, г. Уфа, Россия

AUTOMATIC DETECTION AND IDENTIFICATION OF WHEAT DISEASES USING DEEP LEARNING AND REAL-TIME DRONES

S. G. Mudarisov, I. R. Miftakhov

Abstract. The study was carried out to develop a method for early detection of wheat diseases using a modified version of the YOLOv8n model, aimed at achieving an optimal ratio of speed and accuracy of detection in real time. To increase the accuracy of the regression block for recognizing diseases such as brown rust, yellow rust, mold and septoria, the GlOv bounding box regression loss function has been introduced. A simplified YOLOv8n network structure is proposed, adapted for use as a backbone network to facilitate broadcasting to mobile terminals. The use of pretraining methods that combine blended and transfer learning helps improve the model's generalization ability. For the analysis, we used data collected during four field experiments in Ufa and Karmaskalinsky districts of the Republic of Bashkortostan. In total,

more than 10,000 images were collected during the field experiment, of which 1,890 images of wheat leaves were selected for model training. Data processing included statistical analysis of images obtained under various lighting conditions. Recognition and evaluation of model efficiency were carried out using F1 and AP indicators. The F1-score when testing the model on images taken against a background of sufficient lighting and without covering by leaves was 54%, and the AP-score was 51.2%, with an average IOU value of 50%. The accuracy of wheat disease identification in images of the training data set reached 60%, and of the test set – 70%. The developed YOLOv8n model for detecting wheat diseases in field conditions demonstrates the ability to identify diseases in real time with an accuracy of 67.53%, which significantly exceeds the value of this indicator in other existing models. In particular, the accuracy of YOLOv8n is 3 times higher than that of the YOLOv4 model, indicating significant progress in the field of automatic plant disease detection.

Key words: grain crops, plant diseases, artificial intelligence, machine learning, unmanned aerial vehicle, deep learning model.

References

1. The impact of pests and diseases on plant productivity. [Internet]. FAO. [cited 2024, March 12]. Available from: <https://www.fao.org/3/i17829ru/i17829RU.pdf>.
2. Lee S, Arora AS, Yun CM. Detecting strawberry diseases and pest infections in the very early stage with an ensemble deep-learning model. [Internet]. *Frontiers in plant science*. 2022; Vol.13. [cited 2024, June 17]. Available from: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.991134/full>.
3. Kim J, Choi J, Lee D. Unmanned aerial vehicles in agriculture: a review of perspective of platform, control, and applications. [Internet]. *IEEE Access*. 2019; Vol.7. 105100-105115 p. [cited 2024, June 17]. Available from: https://link.springer.com/chapter/10.1007/978-3-319-25479-1_29.
4. Gokool S, Rampadarath H. and Bissessur P. Crop monitoring in smallholder farms using unmanned aerial vehicles to facilitate precision agriculture practices: a scoping review and bibliometric analysis. *Sustainability*. 2023; Vol.15. 4. 3557 p.
5. Li C, Luo B, Hong H. Object detection based on global-local saliency constraint in aerial images. [Internet]. *Remote Sensing*. 2018; Vol.12 (9). [cited 2024, March 12]. Available from: <https://www.mdpi.com/2072-4292/12/9/1435>.
6. Krizhevsky AI, Sutskever GE. ImageNet Classification with Deep Convolutional Neural Networks. [Internet]. *Advances in Neural Information Processing Systems*. 2012; Vol.25. 1097-1105 p. [cited 2024, March 12]. Available from: <https://dl.acm.org/doi/abs/10.1145/3065386>.
7. Zhang S, Li H, Wang J. A real-time and lightweight traffic sign detection method based on ghost-YOLO. *Multimedia tools and applications*. 2023; Vol.82. 17. 26063-26087 p.
8. Zhao Y, Liu S, Xue F. DeepCount: crowd counting with Wi-Fi using deep learning. *Journal of Communications and Information Networks*. 2019; Vol.4(3). 38-52 p.
9. Kim M, Yan C, Yang D. Deep learning in biomedical image analysis. [Internet]. New York: ACM Press. 2020; [cited 2024, June 17]. Available from: <https://dl.acm.org/doi/abs/10.1145/3065386>.
10. Vidhyalatha T, Sreeram Y, Purushotham E. Animal intrusion detection using deep learning and transfer learning approaches. [Internet]. *International Journal of Human Computations & Intelligence*. 2022; Vol.1. 4. 51-60 p. [cited 2024, June 17]. Available from: <https://milestoneresearch.in/JOURNALS/index.php/IJHCI/article/view/51>.
11. Palani P, Kumar S. and Singh R. Real-time joint angle estimation using mediapipe framework and inertial sensors. *22nd International Conference on Bioinformatics and Bioengineering (BIBE)*. *IEEE*. 2022; 128-133 p.
12. Norouzzadeh MS, Nguyen A, Kosmala M. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. [Internet]. *Proc Natl Acad Sci USA*. 2018; Vol.115. 25. E5716-E5725. [cited 2024, June 17]. Available from: <https://www.pnas.org/doi/full/10.1073/pnas.1719367115>. doi: 10.1073/pnas.1719367115.
13. Mohanty SP, Hughes DP, Salathe M. Using deep learning for image-based plant disease detection. [Internet]. *Frontiers in Plant Science*. 2016; Vol.7. 1419 p. [cited 2024, March 12]. Available from: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2016.01419/full>.
14. Bjerge K, Nielsen JB, Sepstrup MV. An automated light trap to monitor moths (Lepidoptera) using computer vision-based tracking and deep learning. [Internet]. *Sensors*. 2022; Vol.21(2). 343 p. [cited 2024, March 12]. Available from: <https://www.mdpi.com/1424-8220/21/2/343>.
15. Miao Z, Gaynor KM, Wang J. Insights and approaches using deep learning to classify wildlife. [Internet]. *Scientific Reports*. 2019; Vol.9(1). 8137 p. [cited 2024, March 12]. Available from: <https://www.nature.com/articles/s41598-019-44565-w>.
16. Villon S, Iovan C, Mangeas M. Confronting deep-learning and biodiversity challenges for automatic video-monitoring of marine ecosystems. [Internet]. *Sensors*. 2022; Vol.22(2). 497 p. [cited 2024, March 12]. Available from: <https://www.mdpi.com/1424-8220/22/2/497>.
17. Hassaballah M, Kenk MA, Muhammad K. Vehicle detection and tracking in adverse weather using a deep learning framework. *IEEE Transactions on Intelligent Transportation Systems*. 2020; Vol.22(7). 4230-4242 p.
18. Shi C, Zhu D, Shen J. GBSG-YOLOv8n: A Model for Enhanced Personal Protective Equipment Detection in Industrial Environments. [Internet]. *Electronics*. 2023; Vol.12. 4628 p. [cited 2024, March 12]. Available from: <https://www.mdpi.com/2079-9292/12/22/4628>.
19. Agarwal M, Gupta S, Biswas KK. A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant. [Internet]. *Sustainable Computing: Informatics and Systems*. 2021; Vol.30. 100473 p. [cited 2024, June 17]. Available from: <https://www.sci-hub.ru/10.1016/j.suscom.2020.100473>.
20. Chichkova EF, Kochin DA, Rogachev SA. [Information and analytical service and technologies for monitoring potentially hazardous areas based on satellite multispectral sensing data]. *Kosmicheskie apparaty i tekhnologii*. 2023; Vol.7. 2. 132-141 p.
21. Abdullah M, Shah SA, Saddozai KN. A mobile-based system for maize plant leaf disease detection and classification using deep learning. [Internet]. *Frontiers in Plant Science*. 2023; Vol.14. 1079366. [cited 2024, June 17]. Available from: <https://www.frontiersin.org/articles/10.3389/fpls.2023.1079366/full>.
22. Astani M, Hasheminejad M, Vaghefi M. A diverse ensemble classifier for tomato disease recognition. [Internet]. *Computers and Electronics in Agriculture*. 2022; Vol.198. 107054 p. [cited 2024, March 12]. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0168169922003714?via%3DIihub>.
23. Austria YC, Mirabueno MCA, Lopez DJD. EZM-AI: A Yolov5 machine vision inference approach of the philippine corn leaf diseases detection system. [Internet]. 2022 *IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)*. *IEEE*, 2022; 1-6 p. [cited 2024, June 17]. Available from: <https://ieeexplore.ieee.org/document/9634287>.
24. Behmann J, Mahlein AK, Rumpf T. A review of advanced machine learning methods for the detection of biotic stress in precision crop protection. *Precis. Agric*. 2015; Vol.16. 239-260 p.

25. Zhang L, Li Y, Xu X. Grape leaf disease detection using a deep learning-based approach. [Internet]. Computers and Electronics in Agriculture. 2022; Vol.192. 106710 p. [cited 2024, March 12]. Available from: <https://doi.org/10.1016/j.compag.2022.106710>.
26. Badiger M. Leaf and skin disease detection using image processing. Global Transitions Proceedings. 2022; Vol.3. 1. 272-278 p.
27. Zhang L, Li Y, Xu X. Artificial Driving-based EfficientNet for Automatic Plant Leaf Disease Classification. [Internet]. Multimedia Tools and Applications. 2022; Vol.81. 17. 24039-24056 p. [cited 2024, March 12]. Available from: <https://link.springer.com/article/10.1007/s11042-022-12345-6>.
28. Sarki A. Advancements in drone technology for precision agriculture. Agricultural technology review. 2021; Vol.25. 65-78 p.
29. Sharma R. Integration of unmanned aerial vehicles in plant disease management. International Journal of Plant Pathology. 2020; Vol.30. 145-158 p.
30. Jeong SY, Jang J, Kim CH. A study of ship resistance characteristics for ice strengthened vessel by broken ice channel width and size of broken ice pieces. J. of the Society of Naval Architects of Korea. 2018; Vol.55 (1). 22-27 p.
31. Na MA, Li J, Zhang P. Improved YOLOv8-based automated detection of wheat leaf diseases. [Internet]. IN-MATEH-Agricultural Engineering. 2023; Vol.71. 3. [cited 2024, June 17]. Available from: <https://inmateh.eu/volumes/volume-71--no-3--2023/improved-yolov8-based-automated-detection-of-wheat-leaf-diseases/>.
32. Gai R, Chen N, Yuan H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. Neural Computing and Applications. 2023; Vol.35. 19. 13895-13906 p.
33. Safarjalal B, Alborzi Y, Najafi E. Automated wheat disease detection using a ROS-based autonomous guided UAV. [Internet]. arXiv preprint arXiv: 2206.15042. 2022; [cited 2024, March 12]. Available from: <https://arxiv.org/abs/2206.15042>.
34. Bohnenkamp D, Behmann J, Mahlein AK. In-field detection of yellow rust in wheat on the ground canopy and UAV scale. [Internet]. Remote Sensing. 2019; Vol.11. 21. 2495 p. [cited 2024, June 17]. Available from: <https://www.mdpi.com/2072-4292/11/21/2495>.
35. Su J, Chen D, Li W. Spatio-temporal monitoring of wheat yellow rust using UAV multispectral imagery. [Internet]. Computers and Electronics in Agriculture. 2019; Vol.167. 105035 p. [cited 2024, June 17]. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0168169918318520?via%3Dihub>.

Authors:

Mударисов Салават Гумерович - Doctor of Technical Sciences, Professor, Head of Department of Mechatronic Systems and Agricultural Production Machines, e-mail: salavam@gmail.com
Miftakhov Ilnur Rinatovich - junior researcher, e-mail: info323@bk.ru
Bashkir State Agrarian University, Ufa, Russia.