

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Робот для автоматической высадки цветов»
по дисциплине «Проектный практикум»

Заказчик: Шевцов С. А.

Студенты команды TerraTech:
Шестопалов Андрей Сергеевич
Кадочников Арсений Алексеевич
Сайфутдинов Артур Раилович

Екатеринбург, 2025

СОДЕРЖАНИЕ

ЗАДАНИЕ	4
АННОТАЦИЯ.....	5
ВВЕДЕНИЕ.....	6
1 Состав команды.....	8
2 Аналитика	9
2.1 Актуальность	9
2.2 Анализ целевой аудитории	9
2.3 Анализ аналогов	10
2.3.1 FarmBot (США)	10
2.3.2 TerraSentia (США).....	10
2.3.3 Tertill (США)	11
2.3.4 RowBot (Великобритания)	11
2.4 Определение проблемы	11
2.5 Решение проблемы.....	12
2.6 Требования к веб-сервису управления посадками	12
2.6.1 Требования к функциональным характеристикам	12
2.6.2 Требования к надёжности	13
2.6.3 Требования к информационной и программной совместимости....	13
2.7 Требования к работе	14
3 Стек для разработки.....	15
4 Методология разработки	16
5 Распределение задач между разработчиками.....	17
6 Календарный план.....	18
7 Индивидуальные отчёты	19
7.1 Тимлид/Разработчик – Шестопалов А. С.	19
7.2 Аналитик/Разработчик – Кадочников А. А.....	20
7.3 Дизайнер/Разработчик – Сайфутдинов А. Р	21
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24

ПРИЛОЖЕНИЕ А	25
ПРИЛОЖЕНИЕ В	26
ПРИЛОЖЕНИЕ С	27

ЗАДАНИЕ

Разработать робототехническую систему для автоматизированной высадки цветов, включающую 3D-модель робота, схему электроцепей и программное обеспечение. Реализовать веб-сервис для формирования управляющих команд, обеспечив интеграцию с роботом. Провести анализ аналогов, учесть требования к конструкции, и обеспечить устойчивую обработку визуальных данных.

АННОТАЦИЯ

В рамках проекта разработана концепция и функциональная 3D-модель робототехнической системы, предназначеннной для автономной высадки цветов. Одновременно реализован веб-сервис, обеспечивающий формирование управляющих программ для робота.

Проект включал проведение аналитики, проектирование конструкции устройства с учётом требований прочности и технологичности изготовления, а также разработку клиентской и серверной частей программного обеспечения.

Особое внимание удалено обработке возможных ошибок, возникающих при интерпретации визуальных данных и их преобразовании в управляющие команды.

Все этапы работ выполнены в установленные сроки.

ВВЕДЕНИЕ

Современные агротехнологии стремительно развиваются, порождая потребность в автоматизации рутинных процессов, включая посадку растений. Особое внимание уделяется декоративному озеленению и ландшафтному дизайну, где требуется высокая точность, экономия ресурсов и возможность реализации креативных решений. В условиях урбанизации, нехватки трудовых ресурсов и растущего интереса к устойчивому развитию автоматическая высадка растений приобретает особую значимость — от оформления городских общественных пространств до обслуживания ботанических парков и вертикальных ферм. Использование роботизированных систем позволяет не только сократить трудозатраты и минимизировать ошибки, но и создавать уникальные флористические композиции с высокой степенью детализации, недоступной при ручной работе.

Целью данного проекта является разработка автономного робототехнического комплекса, способного в автоматическом режиме производить посадку цветочных растений в заранее определённой области в соответствии с заданным шаблоном или изображением. Основная задача заключается в создании системы, которая сможет распознавать форму посадки, перемещаться по выделенной территории, точно размещать растения в соответствии с координатами шаблона, а также адаптироваться к различным условиям окружающей среды.

Область применения разрабатываемого продукта охватывает как декоративное и муниципальное озеленение, так и частный сектор, агропарки и выставочные площадки. Робот может быть использован в городском благоустройстве, при оформлении парков и скверов, в ландшафтном дизайне частных территорий, а также в образовательных и исследовательских проектах, связанных с робототехникой и автоматизированным сельским хозяйством.

Таким образом, разработка автономного робота для посадки растений представляет собой не только технически интересную, но и социально значимую задачу, способную внести вклад в развитие "умных" городов, устойчивых экосистем и современных ландшафтных решений.

Для достижения цели мы поставили перед собой задачи:

- **Разработать веб-сервис**, позволяющий преобразовывать загруженное изображение в управляющий код для робота с учётом заданных параметров, таких как размеры области посадки и плотность размещения растений;
- **Создать функциональную 3D-модель робота**, демонстрирующую его конструкцию, включая базовую раму, посадочный механизм и систему передвижения;
- **Подготовить подробную инструкцию по сборке и разборке робота**, включающую визуальные схемы, перечень компонентов и последовательность операций;
- **Физическую реализацию прототипа в рамках проекта не предусматривали**, фокус был сделан на цифровом моделировании и программной составляющей, что позволяет использовать результаты как основу для возможной дальнейшей разработки.

1 Состав команды

Шестопалов Андрей Сергеевич – РИ-230913 – тимлид/разработчик;
Кадочников Арсений Алексеевич – РИ-230913 – аналитик/разработчик;
Сайфутдинов Артур Раилович – РИ-230913 – дизайнер/разработчик;

2 Аналитика

2.1 Актуальность

Современные города активно развивают направления, связанные с благоустройством и формированием комфортной городской среды. С каждым годом увеличивается площадь озеленяемых территорий, растут требования к эстетическому качеству и точности оформления цветников, особенно в рамках подготовки к фестивалям, праздникам и общественным мероприятиям.

Одновременно отрасль сталкивается с дефицитом квалифицированной рабочей силы и необходимостью оптимизации рутинных операций. В этих условиях возрастаёт потребность в технологических решениях, способных автоматизировать посадочные работы.

2.2 Анализ целевой аудитории

Целевой аудиторией проекта являются городские и частные службы озеленения, включая специалистов по благоустройству и ландшафтных дизайнеров. Эти пользователи профессионально занимаются проектированием, реализацией и обслуживанием зелёных насаждений в городской среде и на частных территориях. Их деятельность требует профильной подготовки в области агрономии, ландшафтной архитектуры и владения современными средствами автоматизации посадочных работ.

С учётом тенденции к технологизации городской среды, представители данной аудитории демонстрируют высокую восприимчивость к инновационным решениям, таким как роботизированные системы и 3D-печать. Это подтверждается внедрением автоматизированных систем полива, вертикального озеленения и участия в проектах с использованием 3D-печатных конструкций.

2.3 Анализ аналогов

В ходе исследования были рассмотрены несколько современных робототехнических решений, ориентированных на автоматизацию процессов в аграрной и садоводческой сфере.

Наиболее значимыми аналогами, демонстрирующими высокую степень технической зрелости и рыночной востребованности, являются следующие системы:

2.3.1 FarmBot (США)

Открытая автоматизированная платформа для прецизионного земледелия, сочетающая робототехнические механизмы, программное обеспечение и элементы точного земледелия. Обладает широкими функциональными возможностями, включая автоматическую посадку, индивидуальный полив, визуальный мониторинг и прополку. Открытый исходный код и интуитивно понятный интерфейс обеспечивают высокую адаптивность системы к различным пользовательским задачам, включая образовательные цели.

2.3.2 TerraSentia (США)

Компактный автономный робот, предназначенный для автоматизированного сбора фенотипических данных о сельскохозяйственных культурах. Оснащён камерами и LiDAR-сенсорами, способен производить анализ параметров растений и выявлять признаки заболеваний. Отличается автономной навигацией и интеграцией технологий искусственного интеллекта для интерпретации данных, что делает его эффективным инструментом аграрной аналитики.

2.3.3 Tertill (США)

Садовый робот для прополки, функционирующий на солнечной энергии. Предназначен для частных пользователей и не требует подключения к электросети. Робот распознаёт и удаляет сорняки с помощью встроенного триммера, обеспечивая органический уход за участком. Простота в эксплуатации и экологичность делают устройство привлекательным решением для бытового использования в условиях малого садоводства.

2.3.4 RowBot (Великобритания)

Автономная система, ориентированная на эффективное внесение азотных удобрений и посев покровных культур в аграрном производстве. Предназначен для работы в плотных рядах сельскохозяйственных культур, таких как кукуруза. Робот снижает нагрузку на почву и объёмы потребляемых ресурсов, а также способствует уменьшению экологического ущерба за счёт точечного внесения удобрений.

2.4 Определение проблемы

Озеленение городской среды является важнейшей составляющей устойчивого развития мегаполисов, направленной на повышение экологического качества, эстетической привлекательности и комфортности общественных пространств. Однако значительная часть процессов, связанных с высадкой растений, в частности цветов на клумбах, до сих пор выполняется вручную, что создаёт ряд организационно-технологических трудностей.

Наиболее острыми проблемами, с которыми сталкиваются городские и частные службы благоустройства, являются:

- Высокая трудоёмкость работ в периоды пиковой сезонной активности (весна, лето);

- Ограниченные возможности по точному воспроизведению сложных цветочных композиций, требующих аккуратной посадки в соответствии с заранее утверждёнными схемами;
- Влияние человеческого фактора, выражющееся в снижении точности, вариативности исполнения и физической утомляемости персонала;
- Отсутствие универсального технического решения, способного адаптироваться к различным геометриям клумб и индивидуальным требованиям по дизайну посадки.

2.5 Решение проблемы

Предлагается универсальный роботизированный комплекс для автоматизированной высадки растений, предназначенный для работы на клумбах различной формы. Управление осуществляется по координатному коду, сгенерированному веб-сервисом на основе загруженного изображения орнамента.

Робот точно позиционируется на клумбе благодаря системе координатной навигации и механизму пошагового привода, что обеспечивает высокую точность посадки в соответствии с заданной схемой. Совместное использование робота и веб-сервиса позволяет автоматизировать процесс озеленения, сократить трудозатраты и исключить влияние человеческого фактора при формировании цветочных композиций.

2.6 Требования к веб-сервису управления посадками

2.6.1 Требования к функциональным характеристикам

1) Загрузка изображения:

- Пользователь должен иметь возможность загрузить изображение в форматах JPG, PNG, BMP;

2) Обработка изображения:

- Система должна конвертировать изображение в пиксель-арт в соответствие с заданной палитрой;
- Пользователь должен иметь возможность задать площадь посадки и желаемую плотность засадки;
- Применение алгоритма кластеризации и упрощения для адаптации изображения под точечную посадку;

3) Генерация команд управления:

- Возможность корректировки параметров генерации и повторного запуска.
- Возможность скачивания итогового файла.

2.6.2 Требования к надёжности

- 1) Время обработки изображения — не более 5 секунд при стандартном разрешении (до 128×128);
- 2) Генерация команд — не более 3 секунд;
- 3) Обработка файлов до 5 МБ без деградации производительности;
- 4) Непрерывная работа веб-сервиса 24/7 с допустимым временем простоя не более 0,5 % в месяц;

2.6.3 Требования к информационной и программной совместимости

- 1) Поддержка браузеров: Chrome, Firefox, Safari, Microsoft Edge, Яндекс.Браузер (последние две версии);
- 2) Работа на всех распространённых платформах (Windows, macOS, Linux, Android, iOS);
- 3) Передача данных на робота осуществляется в форматах, совместимых с .NET NanoFramework;

2.7 Требования к работе

- 1) Робот должен обладать жесткой несущей рамой, устойчивой к деформациям и вибрациям при движении, обеспечивая точность посадки;
- 2) Конструкция посадочного механизма должна обеспечивать контролируемую посадку без рассыпания и застраивания семян;
- 3) Конструкция робота должна быть модульной, с возможностью адаптации ширины и длины посадочной платформы под различные участки, такие как клумбы, грядки и контейнеры;
- 4) Приводная система должна, обеспечивать движение засевающего механизма по осям X, Y и Z с минимальной погрешностью;
- 5) Система управления должна поддерживать .NET nanoFramework с возможностью работы на микроконтроллерах типа ESP32, STM32 или аналогичных, имеющих достаточные ресурсы для обработки команд и управления периферией;
- 6) Необходимо наличие достаточного количества GPIO-пинов для управления шаговыми двигателями через драйверы, управления двумя сервоприводами посадочного механизма, подключения концевых выключателей на всех осях и дополнительных датчиков;
- 7) Питание должно осуществляться от аккумулятора 12В или 24В, с обязательным включением модулей защиты от перегрузки, короткого замыкания и индикатора уровня заряда;
- 8) Обязательным является наличие концевых выключателей на всех осях для калибровки при запуске (поиска нулевой точки) и предотвращения выхода механизма за пределы рабочей зоны;
- 9) Плата управления и другие чувствительные электронные компоненты должны быть размещены в корпусе с защитой от пыли и влаги, что обеспечит надёжную работу устройства в уличных условиях;

3 Стек для разработки

Исходя требований заказчиков, был выбран такой стек:

- 1) Платформа для управления задачами – Yougile;
- 2) Платформа для создания эскиза приложения – Figma;
- 3) Фронтенд – React JS;
- 4) Бекенд – ASP.NET;
- 5) Среда разработки – JB Rider, Visual Studio Code, Visual Studio;
- 6) Система контроля версий – Git/GitHub;
- 7) 3D моделирование – КОМПАС-3D

4 Методология разработки

В рамках нашего проекта мы приняли решение использовать методологию Agile, поскольку она обеспечивает гибкость, адаптивность и эффективную коммуникацию внутри команды. Такой подход позволяет нам параллельно распределять обязанности между участниками, что ускоряет процесс разработки и снижает риски задержек. Благодаря итеративной модели работы мы можем регулярно получать обратную связь, своевременно вносить изменения и улучшения, а также поддерживать высокий уровень вовлеченности всех участников.

Agile особенно хорошо подходит для нашего проекта, так как мы одновременно разрабатываем сервис и робота, каждый из которых требует скоординированных действий и быстрой реакции на возникающие задачи. Методология позволяет нам вести разработку обеих составляющих параллельно, сохраняя общий вектор развития и обеспечивая прозрачность на всех этапах.

5 Распределение задач между разработчиками

В ходе работы над проектом все разработчики ознакомились с основами построения электрических цепей и базовыми принципами микроэлектроники, поскольку данное направление было для них новым. Это позволило создать общее понимание аппаратной части проекта и повысить качество взаимодействия между командами, работающими над программной и технической составляющими.

Также команда освоила практики совместной разработки с использованием платформы GitHub. Были применены подходы к ведению репозитория, включая ветвление, пулл-реквесты и обязательный code review, что обеспечило контроль качества кода и способствовало коллективному принятию технических решений.

Распределение задач между разработчиками осуществлялось с учётом их текущих навыков, интересов и способности к быстрому обучению. При этом некоторые задачи носили общий характер и выполнялись всеми участниками команды, что способствовало более глубокому погружению каждого в ключевые аспекты проекта.

6 Календарный план

На рисунке 1 показан план работы течении семестра.

		31.03 - 6.04	7.04 - 13.04	14.04 - 20.04	21.04 - 27.04	28.04 - 04.05	05.05 - 11.05	12.05 - 18.05	24.05 - 27.05
Аналитика									
	Определение задач и целей проекта		■						
	Составление требований к работе и сервису		■						
	Анализ целевой аудитории и аналогов		■						
	Составление календарного плана		■	■					
Дизайн									
	Создание эскиза		■	■					
	Создание макета			■					
Разработка сервиса									
	Выбор алгоритма		■						
	Реализация первого контроллера			■					
	Реализация второго контроллера				■	■			
	Генерация кода робота						■	■	
	Создания Фронтиенда		■	■	■				
	Реализация адаптивности						■	■	
Разработка робота									
	Создание прототипа		■	■					
	Создание чертежей				■	■			
	Создание сборочной Инструкции						■	■	
Итог									
	Выкладывание сервиса на хостинг								■

Рисунок 1 – Календарный план

7 Индивидуальные отчёты

7.1 Тимлид/Разработчик – Шестопалов А. С.

В начале работы над проектом я сосредоточился на организационных встречах и решении важных вопросов, связанных с проектом. Я информировал команду о важных датах и событиях, что способствовало эффективному использованию времени и своевременному выполнению задач.

После определения требований заказчика и основной идеи разработки я перешёл к распределению задач между участниками команды. Для этого я создал доску на платформе Yougile. Это позволило нам организовать рабочий процесс и четко обозначить ответственность каждого члена команды. В процессе работы я координировал действия команды, назначал новые задачи и проверял качество их выполнения, выступая основным связующим звеном как внутри команды, так и с куратором.

Пока аналитик определял основные требования к сервису и роботу, а дизайнер составлял эскиз сайта, я занимался разработкой концепции робота.

Также я создал Google Диск для хранения артефактов проекта, что обеспечило удобный доступ к важной информации и способствовало лучшей организации работы.

Одна из моих основных задач, помимо ведения проекта, заключалась в разработке прототипа робота. Я разрабатывал 3D-модели деталей в КОМПАС-3D, создавал технические чертежи с соблюдением ГОСТ стандартов, проектировал схемы электроцепей с учётом энергоэффективности и безопасности системы, а также тщательно подбирал комплектующие, анализируя их стоимость, доступность и совместимость.

Таким образом, моя работа в проекте охватила как организационные, так и технические аспекты. Я приобрел ценный опыт в управлении командой, прототипировании сложных устройств и координации междисциплинарных задач.

7.2 Аналитик/Разработчик – Кадочников А. А.

В рамках проекта по созданию робота для автоматической посадки растений я разработал ключевые программные компоненты системы. Начал с анализа требований, определив необходимую функциональность для робота и веб-сервиса.

Основное внимание уделил разработке алгоритмов обработки изображений. После тестирования различных методов пикселизации выбрал оптимальный подход, сочетающий скорость и качество преобразования. Реализовал алгоритм в виде библиотеки на C#, обеспечивающей генерацию пиксель-арта по заданным параметрам.

Для веб-интерфейса создал ASP.NET контроллер с API для обработки изображений, который полностью покрыл модульными тестами. Затем разработал контроллер для преобразования готового пиксель-арта в команды для робота на nanoFramework. Как результат работы сервиса пользователь получает готовый проект в zip-архиве, который нужно распаковать, запустить с помощью Visual Studio и загрузить код на плату для робота.

Для обеспечения надёжности системы реализовал комплексную обработку ошибок на всех уровнях. Особое внимание уделил валидации входных данных и созданию понятных сообщений об ошибках, что упрощает диагностику проблем в работе системы. Результатом стала отложенная система преобразования изображений в управляющие команды, готовая к интеграции с роботизированной платформой. В ходе проекта я приобрел ценный практический опыт в проектировании и реализации комплексного веб-решения с многоуровневой архитектурой и сложной бизнес-логикой, а также получил знания в сфере робототехники.

7.3 Дизайнер/Разработчик – Сайфутдинов А. Р.

С самого начала проекта я сосредоточился на анализе требований заказчика с сокомандниками, чтобы создать удобный и эстетически привлекательный интерфейс. На основе этого я разработал вариант дизайн-концепции, согласовал его с командой и куратором, после чего приступил к детальной проработке макетов.

Для визуализации идей я использовал Figma, где создал интерактивные прототипы. Это позволило наглядно продемонстрировать логику взаимодействия пользователя с сервисом и оперативно вносить правки на этапе обсуждения. Одновременно я подготовил адаптивные версии дизайна для разных устройств, чтобы обеспечить корректное отображение на всех экранах.

После проектирования интерфейса я занимался фронтенд-разработкой: вёрсткой и интеграцией дизайна в код. Использовал современные технологии, такие как HTML5, CSS3, JavaScript (React.js), чтобы обеспечить высокую производительность и соответствие макетам. Для удобства настроил сборку проекта через Vite.

Все этапы разработки фиксировались в Git, что позволило легко отслеживать изменения и взаимодействовать с backend-разработчиком.

Кроме того, я участвовал в регулярных встречах команды, где презентовал прогресс по дизайну и фронтенду, получал обратную связь и корректировал план работ.

В результате я не только реализовал визуальную часть проекта, но и значительно углубил навыки в проектировании пользовательских сценариев и командной разработке. Этот опыт позволил мне лучше понимать взаимосвязь дизайна и функциональности в digital-продуктах.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы была успешно реализована комплексная робототехническая система, объединяющая механическую часть и программное обеспечение для автоматизированной высадки цветов. Командой была разработана не только функциональная 3D-модель робота, но и веб-сервис, позволяющий формировать управляющие команды на основе визуальных данных, что расширяет возможности применения системы и упрощает её использование.

Проект охватывал широкий спектр задач: от инженерного проектирования до разработки полноценных программных решений. Уделялось внимание как аппаратной надёжности конструкции, так и устойчивости программного обеспечения к ошибкам и сбоям при обработке данных. Все запланированные этапы были успешно выполнены в полном объёме и в установленные сроки, что свидетельствует о высоком уровне организации работы и эффективности командного взаимодействия.

Качество программного продукта было подтверждено результатами тестирования, в ходе которого система продемонстрировала стабильную работу, корректную обработку визуальных данных и надёжность исполнения управляющих команд. Это подтверждает соответствие разработанного решения заявленным требованиям и его готовность к дальнейшему практическому применению.

Полученные в ходе проекта знания и навыки стали важным вкладом в профессиональное развитие участников и создали основу для дальнейшего расширения и масштабирования разработанной системы.

В качестве направлений для будущего развития продукта можно выделить повышение автономности робота за счёт внедрения модулей машинного обучения для распознавания различных типов растений и оптимизации маршрутов передвижения. Также перспективным является добавление возможности удалённого мониторинга и управления через мобильное приложение, что повысит удобство эксплуатации. Развитие

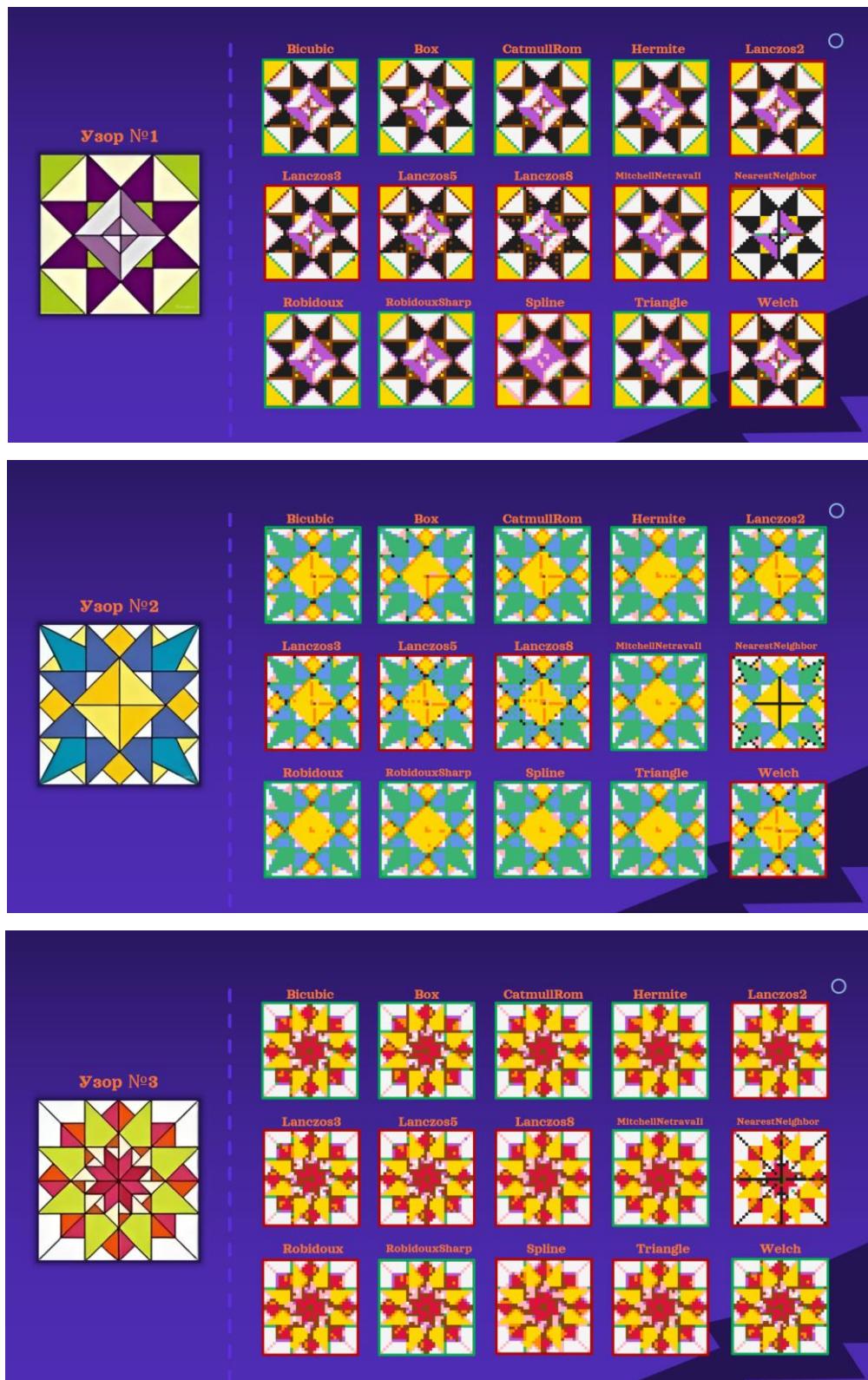
интерфейсов взаимодействия с другими умными системами (например, в рамках концепции «умного сада») позволит интегрировать робот в более широкие экосистемы автоматизированного обслуживания. Всё это создаёт предпосылки для трансформации текущего решения в коммерчески жизнеспособный продукт с возможностью его адаптации под различные сценарии использования в агропромышленном и бытовом секторах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Малышев А.А., Чернов А.И. Микроконтроллеры и микропроцессоры: основы проектирования в системах управления. — СПб.: БХВ-Петербург, 2012. — 448 с.
- 2) Ермаков С.Ю., Глызин С.Д. Основы робототехники: учебное пособие. — М.: Физматлит, 2018. — 256 с.
- 3) Хеккер Ф. Git. Практическое руководство. — СПб.: Питер, 2020. — 320 с.
- 4) Фаулер М. Архитектура корпоративных приложений. — М.: Вильямс, 2016. — 560 с.
- 5) Левин А.В. Основы веб-разработки: HTML, CSS, JavaScript, PHP и MySQL. — М.: Диалектика, 2021. — 384 с.
- 6) Соммервилл И. Инженерия программного обеспечения. — 10-е изд. — СПб.: Питер, 2018. — 800 с.

ПРИЛОЖЕНИЕ А

Результаты обработки изображений разными алгоритмами



ПРИЛОЖЕНИЕ В

Результаты Benchmark-тестирования алгоритмов

Method	Algorithm	Mean	Error	StdDev	Gen0	Gen1	Allocated
ResizeImageWithResampler	Bicubic	225.6 us	1.37 us	1.21 us	4.1504	0.2441	34.14 KB
ResizeImageWithResampler	Box	180.7 us	1.26 us	1.18 us	4.1504	0.2441	33.76 KB
ResizeImageWithResampler	CatmullRom	235.8 us	2.11 us	1.97 us	3.9063	0.4883	34.41 KB
ResizeImageWithResampler	Hermite	238.5 us	1.78 us	1.67 us	3.9063	0.4883	34.26 KB
ResizeImageWithResampler	RobidouxSharp	237.8 us	2.06 us	1.93 us	3.9063	0.4883	34.31 KB

ПРИЛОЖЕНИЕ С

Код класса, тестирующего контроллеры

```
9  public class CodeControllerTests
10 {
11     private WebApplicationFactory<Program> factory;
12     private HttpClient client;
13
14     [SetUp]
15     public void SetUp()
16     {
17         factory = new WebApplicationFactory<Program>();
18         client = factory.CreateClient();
19     }
20
21     [TearDown]
22     public void TearDown()
23     {
24         client.Dispose();
25         factory.Dispose();
26     }
27
28     [Test]
29     public async Task GenerateCodeReturnsBinaryFileWhenValidRequest()
30     {
31         var path = Path.Combine(Directory.GetCurrentDirectory(), "../../TestAssets/sample.png");
32         var binPath = Path.Combine(Directory.GetCurrentDirectory(), "Resources/robot_code.bin");
33
34         Directory.CreateDirectory(Path.GetDirectoryName(binPath)!!);
35         await File.WriteAllBytesAsync(binPath, new byte[] { 0x01, 0x02, 0x03 });
36
37         using var imageStream = File.OpenRead(path);
38         using var content = new MultipartFormDataContent();
39
40         content.Add(new StreamContent(imageStream)
41         {
42             Headers = { ContentType = new MediaTypeHeaderValue("image/png") }
43         }, "image", "sample.png");
44
45         content.Add(new StringContent("100"), "length");
46         content.Add(new StringContent("100"), "width");
47
48         var response = await client.PostAsync("/api/code/generate", content);
49
50         Assert.That(response.StatusCode, Is.EqualTo(HttpStatusCode.OK));
51         Assert.That(response.Content.Headers.ContentType?.MediaType, Is.EqualTo("application/octet-stream"));
52         Assert.That(response.Content.Headers.ContentDisposition?.FileName, Is.EqualTo("robot_code.bin"));
53     }
54
55     [Test]
56     & Virtual
57     public async Task GenerateCodeReturnsBadRequestWhenImageIsMissing()
58     {
59         using var content = new MultipartFormDataContent();
60         content.Add(new StringContent("100"), { name: "length" });
61         content.Add(new StringContent("100"), { name: "width" });
62
63         var response = await client.PostAsync( requestUri: "/api/code/generate", content);
64
65         Assert.That(response.StatusCode, expression: Is.EqualTo( expected: HttpStatusCode.BadRequest));
66     }
67
68     [Test]
69     & Virtual
70     public async Task GenerateCodeReturnsUnsupportedMediaWhenImageTypeIsInvalid()
71     {
72         using var stream = new MemoryStream( buffer: System.Text.Encoding.UTF8.GetBytes( @"invalid"));
73         using var content = new MultipartFormDataContent();
74
75         content.Add(new StreamContent(stream)
76         {
77             Headers = { ContentType = new MediaTypeHeaderValue("text/plain") }
78         }, { name: "image", fileName: "text.txt" });
79
80         content.Add(new StringContent("100"), { name: "length" });
81         content.Add(new StringContent("100"), { name: "width" });
82
83         var response = await client.PostAsync( requestUri: "/api/code/generate", content);
84
85         Assert.That(response.StatusCode, expression: Is.EqualTo( expected: HttpStatusCode.UnsupportedMediaType));
86     }

```