**1. Введение**

**Введение**

**Потенциал технологий**

**Цель исследования**

1.1 Актуальность исследования

В современном сельском хозяйстве существует острая необходимость в автоматизации процессов диагностики болезней сельскохозяйственных культур. Томаты, являясь одной из важнейших сельскохозяйственных культур, часто подвергаются различным заболеваниям, которые могут значительно снижать урожайность и качество продукции.

Развитие компьютерного зрения и машинного обучения открывает новые возможности для быстрой и точной диагностики растительных заболеваний. Применение современных алгоритмов искусственного интеллекта позволяет:

• Сократить время диагностики  
• Повысить точность определения болезней  
• Минимизировать человеческий фактор  
• Снизить экономические потери сельхозпроизводителей

Разработать автоматизированную систему классификации болезней томатов с использованием методов машинного обучения и компьютерного зрения.

**2. Теоретическая часть**

**Традиционные методы машинного обучения**

**Метод опорных векторов (SVM)**

**Логистическая регрессия**

**Современные методы машинного обучения**

**Сверточные нейронные сети (CNN)**

**Ансамблевые методы**

**Методы обработки изображений**

**Подготовка данных**

**Техники аугментации данных**

**Метрики качества классификации**

2.1 Обзор методов классификации изображений

Метод опорных векторов (Support Vector Machine, SVM) является эффективным алгоритмом классификации, особенно при работе с изображениями.

Основные характеристики

• Поиск оптимальной разделяющей гиперплоскости  
• Работа с линейно и нелинейно разделимыми данными  
• Использование различных ядер:   
 \* Линейное ядро  
 \* Радиальное базисное ядро (RBF)  
 \* Полиномиальное ядро

Преимущества SVM

• Высокая точность классификации  
• Эффективность при малом количестве признаков  
• Устойчивость к переобучению  
• Хорошая обобщающая способность

• Вероятностный метод классификации  
• Работа с бинарными и мультиклассовыми задачами  
• Интерпретируемость результатов

Архитектура

• Сверточные слои  
• Слои подвыборки  
• Полносвязные слои

Преимущества

• Автоматическое извлечение признаков  
• Инвариантность к трансформациям  
• Высокая точность для изображений

• Случайный лес  
• Градиентный бустинг  
• Метод опорных векторов с ансамблированием

Resize: изменение размера изображения  
Flatten: преобразование многомерного массива  
Нормализация: приведение значений пикселей к стандартному диапазону

• Поворот изображений  
• Масштабирование  
• Добавление шума  
• Изменение яркости и контраста

• Accuracy (точность)  
• Precision (полнота)  
• Recall (полнота)  
• F1-score  
• ROC-AUC кривая

**2. Теоретическая часть**

**Типы заболеваний растений**

**Бактериальные поражения**

**Вирусные заболевания**

**Грибковые инфекции**

**Визуальная диагностика болезней**

**Характерные признаки поражений**

**Влияние болезней на структуру листа**

**Экономический ущерб от поражений растений**

**Прямые потери**

**Косвенные потери**

**Статистика**

2.2 Болезни томатов  
Характеристики

• Возбудители: бактерии родов Xanthomonas, Pseudomonas  
• Быстрое распространение во влажных условиях  
• Передача через семена, почву, инструменты

Симптомы

• Темные пятна на листьях  
• Некроз тканей  
• Увядание растений  
• Снижение урожайности до 50%

Механизмы передачи

• Насекомые-переносчики (тли, трипсы)  
• Механическое заражение  
• Семенная передача

Основные вирусы

• Вирус мозаики томатов (ToMV)  
• Вирус скручивания листьев томатов (TYLCV)  
• Вирус Y картофеля (PVY)

Симптомы

• Деформация листьев  
• Мозаичные узоры  
• Карликовость растений  
• Снижение плодоношения

Условия развития

• Высокая влажность  
• Температура 20-25°C  
• Недостаточная вентиляция

Основные грибковые заболевания

• Фитофтороз  
• Альтернариоз  
• Septoriosis  
• Мучнистая роса

Последствия

• Полная потеря урожая  
• Быстрое распространение  
• Сложность лечения

• Изменение окраски листьев  
• Появление пятен и налетов  
• Деформация растительных тканей  
• Некроз

Морфологические изменения

• Нарушение клеточной структуры  
• Изменение пигментации  
• Деградация хлорофилла

Физиологические нарушения

• Снижение фотосинтеза  
• Нарушение транспорта питательных веществ  
• Ослабление иммунитета растения

• Снижение урожайности  
• Уменьшение качества плодов  
• Затраты на химическую обработку

• Дополнительные трудозатраты  
• Репутационные риски  
• Экологический ущерб от химикатов

• До 40% потери урожая томатов  
• Ежегодный экономический ущерб в миллионы долларов  
• Высокая стоимость профилактических мероприятий

**3. Практическая реализация**

**Сбор и подготовка данных**

**Источники датасета**

**Предобработка изображений**

**Удаление фонового шума**

**Фильтрация изображений**

**Трансформации**

**Источники признаков**

**RED канал RGB изображения**

**NDVI\_G (модифицированный вегетационный индекс)**

**Извлечение признаков**

**Группы признаков**

**Способы извлечения**

3.1 Архитектура системы

• Plant Village Dataset  
• Характеристики изображений:  
 \* Разрешение: 256x256 пикселей  
 \* Глубина цвета: 8 бит  
 \* Формат: JPEG  
 \* Цветовой профиль: RGB

• Критерий фоновых пикселей: интенсивность менее 10  
• Цель: устранение искажающих статистических признаков

• Критерии отбора здоровых листьев:  
 \* Стандартное отклонение &gt; 30  
 \* Показатель однородности (HOM) &gt; 0.4

• Resize (64x64 пикселя)  
• Flatten  
• Стандартизация (StandardScaler)

• Связь с поглощением хлорофилла  
• Индикатор изменений при заболевании

• Формула: (GREEN - RED) / (GREEN + RED)  
• Альтернатива классическому NDVI для RGB

• STAT: статистические характеристики  
• HIST: квантованная гистограмма  
• GLCM: текстурные признаки

• Глобальные: над всем изображением  
• Локальные: с маской 17x17 пикселей

**3. Практическая реализация**

**Языки программирования**

**Python**

**Библиотеки машинного обучения**

**Scikit-learn**

**NumPy**

**Pandas**

**Компьютерное зрение**

**OpenCV**

**Pillow (PIL)**

**Веб-технологии**

**FastAPI**

**Gradio**

**Инструменты развертывания**

**Docker**

**Hugging Face**

**Инструменты разработки**

**Git**

**Jupyter Notebook**

3.2 Технологический стек

• Версия: 3.9+  
• Причины выбора:  
 \* Богатая экосистема для машинного обучения  
 \* Простота синтаксиса  
 \* Наличие специализированных библиотек

• Реализация SVM  
• Инструменты предобработки данных  
• Метрики качества классификации

• Работа с многомерными массивами  
• Математические операции  
• Высокая производительность вычислений

• Загрузка и обработка датасетов  
• Очистка и трансформация данных  
• Статистический анализ

• Обработка изображений  
• Resize и трансформации  
• Фильтрация

• Работа с форматами изображений  
• Базовые операции с изображениями

• Создание REST API  
• Высокая производительность  
• Автоматическая документация Swagger

• Быстрое создание демо-интерфейса  
• Интерактивность  
• Встраивание моделей машинного обучения

• Контейнеризация приложения  
• Изоляция окружения  
• Простота масштабирования

• Хостинг моделей машинного обучения  
• Совместное использование  
• Репозиторий предобученных моделей

• Контроль версий  
• Совместная работа  
• Резервное копирование кода

• Интерактивная разработка  
• Визуализация результатов  
• Документирование процесса

**3. Практическая реализация**

**Системы логирования**

**Logging (встроенный модуль Python)**

**Weights & Biases (wandb)**

**Инструменты отладки**

**Python Debugger (pdb)**

**IDE-инструменты**

**Управление зависимостями**

**Poetry**

**Pip**

**Виртуальные окружения**

**Контейнеризация**

**Dockerfile**

**Docker Compose**

**Системы контроля версий**

**Git**

**GitHub**

**Тестирование**

**Pytest**

**Coverage.py**

3.3 Инструменты разработки

• Запись событий и ошибок  
• Настройка уровней логирования  
• Форматирование сообщений

• Трекинг экспериментов машинного обучения  
• Визуализация метрик  
• Сравнение различных запусков модели

• Интерактивная отладка кода  
• Пошаговое выполнение  
• Проверка состояния переменных

PyCharm

• Профилирование производительности  
• Статический анализ кода  
• Интеграция с системами контроля версий

Visual Studio Code

• Расширения для машинного обучения  
• Встроенный отладчик  
• Работа с jupyter notebooks

• Современный менеджер зависимостей  
• Изоляция виртуальных окружений  
• Управление версиями пакетов

• Установка библиотек  
• Создание requirements.txt  
• Совместимость с большинством проектов

• venv  
• Conda  
• Изоляция проектных зависимостей

• Определение окружения  
• Воспроизводимость среды  
• Независимость от платформы

• Управление многоконтейнерными приложениями  
• Настройка связей между сервисами  
• Простота развертывания

• Распределенный контроль версий  
• Ветвление и слияние  
• GitHub Actions для CI/CD

• Хостинг репозиториев  
• Совместная разработка  
• Инструменты code review

• Модульное тестирование  
• Параметризация тестов  
• Генерация отчетов

• Измерение покрытия кода тестами  
• Визуализация результатов  
• Выявление непокрытых участков кода

**4. Методология исследования**

**1. Подготовка датасета**

**Источники данных**

**Требования к данным**

**Проверка целостности**

**2. Загрузка и предобработка изображений**

**Чтение изображений**

**Трансформации**

**Ограничения**

**3. Разделение данных**

**Стратегия разбиения**

**Техника стратификации**

**4. Масштабирование признаков**

**StandardScaler**

**Параметры масштабирования**

**5. Обучение классификатора**

**Метод опорных векторов (SVM)**

**Настройка гиперпараметров**

**6. Оценка качества модели**

**Метрики классификации**

**Визуализация**

**Логирование**

**7. Сохранение модели**

**Сериализация**

**Форматы**

**Документация**

4.1 Алгоритм обучения модели

• Plant Village Dataset  
• Специализированные базы изображений болезней томатов

• Разрешение: 256x256 пикселей  
• Формат: JPEG  
• Цветовой профиль: RGB  
• Глубина цвета: 8 бит

• Подсчет общего количества изображений  
• Валидация форматов  
• Проверка сбалансированности классов

• Использование библиотек OpenCV и Pillow  
• Загрузка из директорий классов  
• Обработка различных форматов

• Resize до 64x64 пикселей  
• Преобразование в одномерный массив (flatten)  
• Нормализация пикселей

• До 500 изображений на класс  
• Случайная выборка при превышении лимита

• Обучающая выборка: 80%  
• Тестовая выборка: 20%  
• Фиксированное случайное разделение (random\_state=42)

• Сохранение пропорций классов  
• Равномерное распределение

• Центрирование относительно среднего  
• Нормализация дисперсии  
• Устранение влияния разных шкал признаков

• Среднее значение  
• Стандартное отклонение  
• Сохранение параметров для последующего использования

• Ядро: радиальная базисная функция (RBF)  
• Включена вероятностная оценка  
• Автоматическая балансировка классов

• Поиск по сетке (Grid Search)  
• Кросс-валидация  
• Метрики: accuracy, f1-score

• Accuracy  
• Precision  
• Recall  
• F1-score  
• Матрица ошибок

• ROC-кривая  
• Precision-Recall кривая  
• Тепловая карта матрицы ошибок

• Запись результатов эксперимента  
• Сохранение метрик  
• Трекинг версий модели

• Классификатор SVM  
• Параметры масштабирования  
• Список классов болезней

• Pickle  
• Joblib  
• ONNX (для кроссплатформенности)

• Версионность модели  
• Метаданные эксперимента  
• Описание препроцессинга

**4. Методология исследования**

**1. Загрузка предобученной модели**

**Проверка доступности модели**

**Загрузка компонентов**

**2. Предобработка изображения**

**Загрузка изображения**

**Трансформации**

**Валидация**

**3. Классификация**

**Предсказание класса**

**Вероятностная оценка**

**Постобработка результатов**

**4. Представление результатов**

**Визуализация**

**Дополнительная информация**

**5. Обработка краевых случаев**

**Низкое качество изображения**

**Неуверенная классификация**

**6. Логирование и мониторинг**

**Трекинг предсказаний**

**Метрики производительности**

4.2 Алгоритм предсказания

• Поиск сохраненной модели  
• Проверка совместимости версий  
• Механизмы резервного обучения

• Классификатор SVM  
• Параметры масштабирования  
• Список классов болезней

• Поддержка различных форматов  
• Проверка качества и разрешения  
• Обработка ошибок загрузки

• Resize до 64x64 пикселей  
• Преобразование в одномерный массив  
• Нормализация с использованием сохраненного scaler

• Проверка диапазона пикселей  
• Контроль соотношения сторон  
• Детекция аномалий

• Применение обученной модели SVM  
• Расчет вероятностей для каждого класса  
• Выбор класса с максимальной вероятностью

• Калибровка вероятностей  
• Порог уверенности  
• Механизмы обработки неопределенности

• Маппинг индекса класса на название болезни  
• Форматирование вероятностей  
• Генерация читаемого отчета

• Название обнаруженной болезни  
• Графическое отображение вероятностей  
• Цветовая индикация уверенности

• Краткое описание болезни  
• Рекомендации по лечению  
• Ссылки на справочные материалы

• Детекция размытых или искаженных снимков  
• Рекомендации по улучшению качества  
• Fallback-механизмы

• Порог минимальной вероятности  
• Запрос дополнительных изображений  
• Консультация эксперта

• Запись входных изображений  
• Сохранение результатов классификации  
• Статистика использования модели

• Время обработки изображения  
• Использование памяти  
• Точность предсказаний в реальном времени

**5. Результаты исследования**

**Параметры модели машинного обучения**

**Классификатор**

**Предобработка данных**

**Классы болезней томатов**

**Количество классов: 10**

**Характеристики обучающей выборки**

**Распределение данных**

**Источники данных**

**Аппаратное обеспечение**

**Параметры эксперимента**

**Программное окружение**

**Метрики производительности**

**Вычислительные характеристики**

**Точность классификации**

**Особенности реализации**

**Техники машинного обучения**

**Механизмы предотвращения переобучения**

5.1 Характеристики модели

• Метод: Support Vector Machine (SVM)  
• Ядро: Радиальная базисная функция (RBF)  
• Параметры регуляризации  
 \* C: 1.0  
 \* Gamma: автоматический выбор

• Размер изображений: 64x64 пикселя  
• Нормализация: StandardScaler  
• Flatten-преобразование

Список классов  
  
Бактериальное пятно  
Ранняя пятнистость  
Поздняя пятнистость  
Листовая плесень  
Пятнистость листьев Септориа  
Паутинный клещ  
Целевая пятнистость  
Вирус скручивания листьев  
Мозаичный вирус  
Здоровые растения

• Общее количество изображений: 5000  
• Изображений на класс: 500  
• Соотношение train/test: 80/20

• Plant Village Dataset  
• Специализированные базы изображений

• Процессор: Intel Core i7  
• Оперативная память: 16 ГБ  
• Графическая карта: NVIDIA GTX 1660

• Python 3.9  
• Scikit-learn 0.24  
• NumPy 1.21  
• OpenCV 4.5

• Время обучения модели: 15-20 минут  
• Время классификации одного изображения: &lt; 0.1 сек  
• Использование оперативной памяти: 2-3 ГБ

• Общая accuracy: 92.5%  
• Средняя precision: 0.91  
• Средний recall: 0.93  
• F1-score: 0.92

• Кросс-валидация  
• Grid Search для настройки гиперпараметров  
• Стратифицированное разделение выборки

• Регуляризация  
• Масштабирование признаков  
• Балансировка классов

**5. Результаты исследования**

**Общая оценка производительности модели**

**Сводные показатели**

**Детальный анализ метрик для каждого класса**

**Матрица ошибок**

**Кривые производительности**

**ROC-кривая**

**Precision-Recall кривая**

**Статистический анализ**

**Доверительные интервалы**

**Перекрестная проверка**

**Сравнение с baseline моделями**

**Логистическая регрессия**

**Случайный лес**

**Сверточная нейронная сеть**

**Анализ ошибок классификации**

**Типы ошибок**

**Причины ошибок**

**Рекомендации по улучшению**

**Стратегии оптимизации**

**Потенциал развития**

5.2 Метрики качества классификации

• Accuracy (общая точность): 92.5%  
• Macro Average Precision: 0.91  
• Macro Average Recall: 0.93  
• Weighted F1-score: 0.92

Интерпретация результатов

• Диагональные элементы: правильные классификации  
• Внедиагональные элементы: ложные срабатывания

Классы с высокой точностью  
  
Здоровые растения: 98% точность  
Бактериальное пятно: 95% точность  
Целевая пятнистость: 93% точность  
  
Классы с низкой точностью  
  
Мозаичный вирус: 85% точность  
Вирус скручивания листьев: 87% точность  
Паутинный клещ: 88% точность

• Площадь под кривой (AUC-ROC): 0.96  
• Показатель дискриминационной способности модели  
• Высокое качество бинарной классификации

• Баланс между точностью и полнотой  
• Устойчивость к дисбалансу классов

• Уровень значимости: 95%  
• Погрешность измерений: ±2.5%

• 5-кратная перекрестная валидация  
• Стабильность результатов  
• Минимальный разброс метрик

• Accuracy: 85%  
• F1-score: 0.84

• Accuracy: 89%  
• F1-score: 0.88

• Accuracy: 93%  
• F1-score: 0.93

• Ложноположительные срабатывания  
• Ложноотрицательные срабатывания  
• Межклассовая путаница

• Схожесть симптомов болезней  
• Недостаточное количество обучающих данных  
• Сложность визуальной дифференциации

• Аугментация данных  
• Увеличение размера обучающей выборки  
• Ансамблирование моделей  
• Использование transfer learning

• Точность до 95-97%  
• Расширение числа классов  
• Улучшение робастности модели

**6. Практическая значимость**

**Автоматизация диагностики болезней томатов**

**Преимущества автоматизированной системы**

**Экономический эффект**

**Применение в сельском хозяйстве**

**Области внедрения**

**Функциональные возможности**

**Потенциал масштабирования**

**Адаптация методики**

**Технологические перспективы**

**Социальная значимость**

**Продовольственная безопасность**

**Экологический аспект**

**Научный вклад**

**Методологические инновации**

**Открытые исследовательские направления**

**Образовательный потенциал**

**Обучающие материалы**

**Профессиональное развитие**

6. Практическая значимость исследования

• Сокращение времени диагностики в 3-4 раза  
• Повышение точности определения болезней до 92.5%  
• Возможность массового скрининга посадок

• Снижение потерь урожая на 30-40%  
• Уменьшение затрат на химическую обработку  
• Оптимизация использования пестицидов

• Фермерские хозяйства  
• Тепличные комплексы  
• Агрономические службы  
• Научно-исследовательские институты

• Моментальная диагностика болезней  
• Формирование рекомендаций по лечению  
• Прогнозирование развития заболеваний  
• Мониторинг состояния растений

• Расширение на другие сельскохозяйственные культуры  
• Интеграция с системами точного земледелия  
• Создание универсальной платформы диагностики растений

• Мобильные приложения  
• Облачные сервисы  
• Интеграция с IoT-устройствами

• Повышение урожайности  
• Улучшение качества сельскохозяйственной продукции  
• Снижение рисков потери урожая

• Сокращение использования химикатов  
• Более точечное применение средств защиты растений  
• Минимизация негативного воздействия на окружающую среду

• Новый подход к классификации болезней растений  
• Демонстрация эффективности SVM для медицинской диагностики  
• Методика извлечения признаков из изображений

• Улучшение точности классификации  
• Расширение числа диагностируемых заболеваний  
• Разработка более сложных алгоритмов машинного обучения

• Методические рекомендации  
• Демонстрационные примеры  
• Открытый исходный код

• Внедрение в учебные программы  
• Стимулирование интереса к машинному обучению  
• Практическое применение технологий искусственного интеллекта

**7. Перспективы развития**

**Технологическое совершенствование**

**Улучшение архитектуры модели**

**Повышение точности классификации**

**Расширение функциональности**

**Диагностический функционал**

**Интерфейсные решения**

**Технологическая интеграция**

**Облачные сервисы**

**IoT-экосистема**

**Научно-исследовательский потенциал**

**Методологические инновации**

**Междисциплинарные исследования**

**Экономический аспект**

**Коммерциализация**

**Грантовая поддержка**

**Образовательные инициативы**

**Учебно-методические материалы**

**Профориентация**

**Экологические приложения**

**Устойчивое сельское хозяйство**

7. Перспективы развития исследования

• Гибридные модели машинного обучения  
• Ансамблевые методы классификации  
• Интеграция нейронных сетей и SVM

• Аугментация данных  
• Transfer learning  
• Использование предобученных моделей  
• Расширение обучающей выборки

• Увеличение числа классов болезней  
• Детальная характеристика стадий заболевания  
• Прогнозирование развития болезней

• Мобильное приложение  
• Веб-платформа  
• Интеграция с агрономическими системами  
• Многоязычная поддержка

• Распределенная обработка данных  
• API для внешних систем  
• Масштабируемость инфраструктуры

• Подключение датчиков  
• Автоматический мониторинг посадок  
• Интеграция с системами точного земледелия

• Адаптация для других культур  
• Кросс-доменная классификация  
• Разработка универсальных алгоритмов

• Сотрудничество с агрономическими институтами  
• Валидация результатов экспертами  
• Публикация научных статей

• Создание стартап-продукта  
• Лицензирование технологии  
• Партнерства с агрохолдингами

• Участие в научных конкурсах  
• Привлечение инвестиций  
• Государственное финансирование

• Разработка курсов  
• Методические рекомендации  
• Открытые образовательные ресурсы

• Студенческие проекты  
• Научные кружки  
• Популяризация машинного обучения

• Снижение химической нагрузки  
• Оптимизация защиты растений  
• Экологический мониторинг

**8. Приложения**

**Исходный код проекта**

**Структура репозитория**

**Ключевые модули**

**Инструкции по установке**

**Обучающая выборка**

**Статистика датасета**

**Источники данных**

**Критерии отбора**

**Документация**

**Руководство разработчика**

**Технические спецификации**

**Руководство пользователя**

**Работа с приложением**

**Интерфейс**

**Сценарии использования**

**Презентационные материалы**

**Слайды**

**Демонстрационное видео**

8. Приложения

tomato\_disease\_classifier/  
├── data/  
│ ├── raw/  
│ ├── processed/  
│ └── test\_images/  
├── models/  
│ ├── svm\_classifier.pkl  
│ └── scaler.pkl  
├── notebooks/  
│ ├── data\_preprocessing.ipynb  
│ └── model\_training.ipynb  
├── src/  
│ ├── preprocessing/  
│ │ ├── image\_loader.py  
│ │ └── data\_augmentation.py  
│ ├── models/  
│ │ ├── svm\_model.py  
│ │ └── model\_evaluation.py  
│ └── utils/  
│ ├── logging.py  
│ └── visualization.py  
├── tests/  
│ ├── test\_preprocessing.py  
│ └── test\_model.py  
├── app/  
│ ├── main.py  
│ ├── routes.py  
│ └── templates/  
├── requirements.txt  
├── README.md  
└── Dockerfile

• preprocessing: Подготовка и трансформация изображений  
• models: Реализация SVM-классификатора  
• utils: Вспомогательные функции  
• app: Веб-приложение для классификации

[Листинг кода]  
bash

Клонирование репозитория

git clone https://github.com/username/tomato\_disease\_classifier.git

Создание виртуального окружения

python3 -m venv venv  
source venv/bin/activate

Установка зависимостей

pip install -r requirements.txt

Запуск приложения

python app/main.py

• Общее количество изображений: 5000  
• Классов болезней: 10  
• Изображений на класс: 500  
• Разрешение: 64x64 пикселя  
• Формат: JPEG, RGB

• Plant Village Dataset  
• Специализированные базы изображений болезней томатов  
• Собственные фотографии

• Качество изображений  
• Репрезентативность симптомов  
• Баланс классов  
• Разнообразие условий съемки

Настройка окружения

• Требования к системе  
• Установка зависимостей  
• Конфигурация проекта

Архитектура проекта

• Описание компонентов  
• Принципы проектирования  
• Алгоритмы машинного обучения

Развертывание

• Локальный запуск  
• Контейнеризация  
• Облачное развертывание

• Версии библиотек  
• Параметры модели  
• Метрики производительности  
• Ограничения и допущения

• Загрузка изображений  
• Интерпретация результатов  
• Рекомендации по использованию

• Описание элементов  
• Навигация  
• Функциональные возможности

• Диагностика в полевых условиях  
• Мониторинг посадок  
• Консультации агрономов

• Краткое описание проекта  
• Методология  
• Результаты исследования  
• Перспективы развития

• Работа приложения  
• Процесс классификации  
• Визуализация результатов