

Повышение эффективности управления микроклиматом теплиц с использованием нейросетевых технологий и распределённых встраиваемых систем

В.А. Езунов¹, А.А. Кузьменко¹, А.С. Борисов¹, О.А. Старовойтова²

¹Волгоградский государственный технический университет

²Санкт-Петербургский государственный университет аэрокосмического приборостроения

Аннотация: В данной статье рассматриваются особенности разработки и применения нейроподобной автоматизированной системы управления микроклиматом тепличного комплекса. Актуальность темы обусловлена необходимостью повышения устойчивости сельскохозяйственного производства в условиях роста населения и климатических изменений. Представлена архитектура интеллектуальной системы, включающая иерархическую сенсорную сеть на базе микроконтроллеров STM32, модуль анализа на Raspberry Pi 5 с нейросетевой обработкой данных и распределённый исполнительный уровень. Описаны методы обработки телеметрии, алгоритмы адресации и опроса устройств, а также нейросетевые модели (YOLO и EfficientNetB3) для диагностики состояния растений. Проведён эксперимент по классификации заболеваний растений, продемонстрировавший высокую точность и возможность применения системы в режиме реального времени.

Ключевые слова: автоматизация теплиц, микроклимат, нейросетевой анализ, STM32, Raspberry Pi, глубокое обучение, YOLO, EfficientNet, IoT, классификация заболеваний растений.

Введение. Актуальность автоматизации теплиц и управления микроклиматом обусловлена прогнозируемым ростом населения до 9,7 млрд к 2050 г. [1] и усиливающимся воздействием климатических изменений на аграрное производство [2]. Традиционные системы управления, основанные на линейных моделях или эмпирических правилах, слабо адаптируются к внешним возмущениям, что приводит к ошибкам до 15–20 %, снижению урожайности и перерасходу энергии [3]. Автоматизированные комплексы всё шире применяются для снижения зависимости от ручного труда в потенциально опасных условиях [4] и находят применение в различных сферах — от сельского хозяйства до медицины и обороны [5].

Метод нелинейного прогнозирующего управления (NMPC) улучшает стабилизацию за счёт учёта будущих возмущений, однако его

вычислительная сложность ограничивает применение в реальном времени [6]. Внедрение IoT и облачных технологий позволяет оперативно собирать данные (температура, влажность, CO₂ и др.) и использовать прогнозные алгоритмы, снижая энергозатраты на 15–25 % и уменьшая необходимость постоянного мониторинга [7]. Глубокие нейронные сети успешно выявляют нелинейные зависимости между параметрами микроклимата и ростом растений, обеспечивая точность прогноза температуры до 1,5 °С против 5 °С у классических моделей, при меньших вычислительных затратах [8]. Гибридные системы, объединяющие IoT, нейросетевые методы и глубокое усиленное обучение (DRL), демонстрируют более эффективное и энергосберегающее управление микроклиматом по сравнению с традиционными MPC и чистыми RL-подходами [9].

Архитектура. Разрабатываемая система реализует нейроподобную архитектуру распределённого мониторинга и управления. В её основе лежит трёхкомпонентная модель, состоящая из чувствительной, анализирующей и исполнительной частей (рисунок 1).

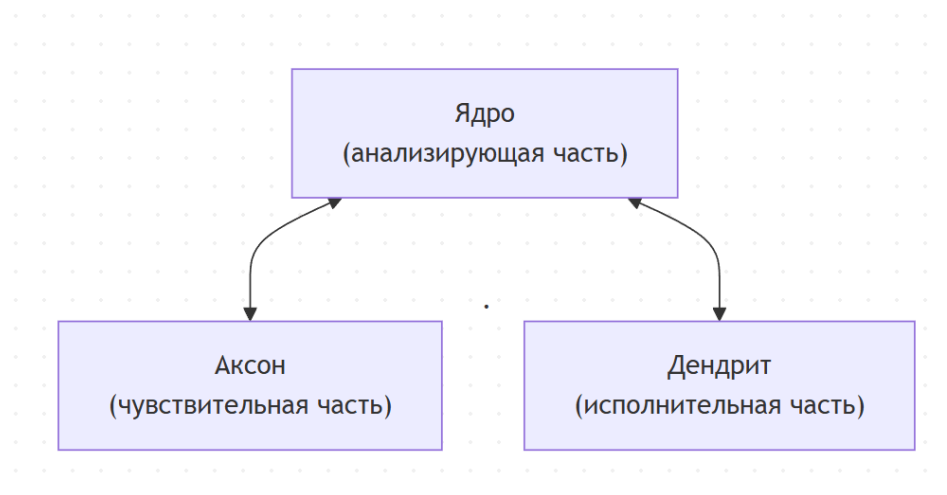


Рисунок 1. – Архитектура информационной системы

Анализирующая часть (ядро) отвечает за прием телеметрических данных, их обработку с помощью нейросетевых алгоритмов и выработку управляющих сигналов. Чувствительная часть (аксон) обеспечивает сбор

полевых данных и передачу их в ядро. Исполнительная часть (дендрит) распределённо доставляет сформированные ядром команды к приводам и исполнительным механизмам.

Чувствительная часть («аксон нейрона»). Чувствительная часть реализована в виде двухуровневой иерархии: высокоуровневого управляющего узла на базе STM32H723VGT6 и множества низкоуровневых сенсорных модулей на STM32G0B1CBT6 (рисунок 2)

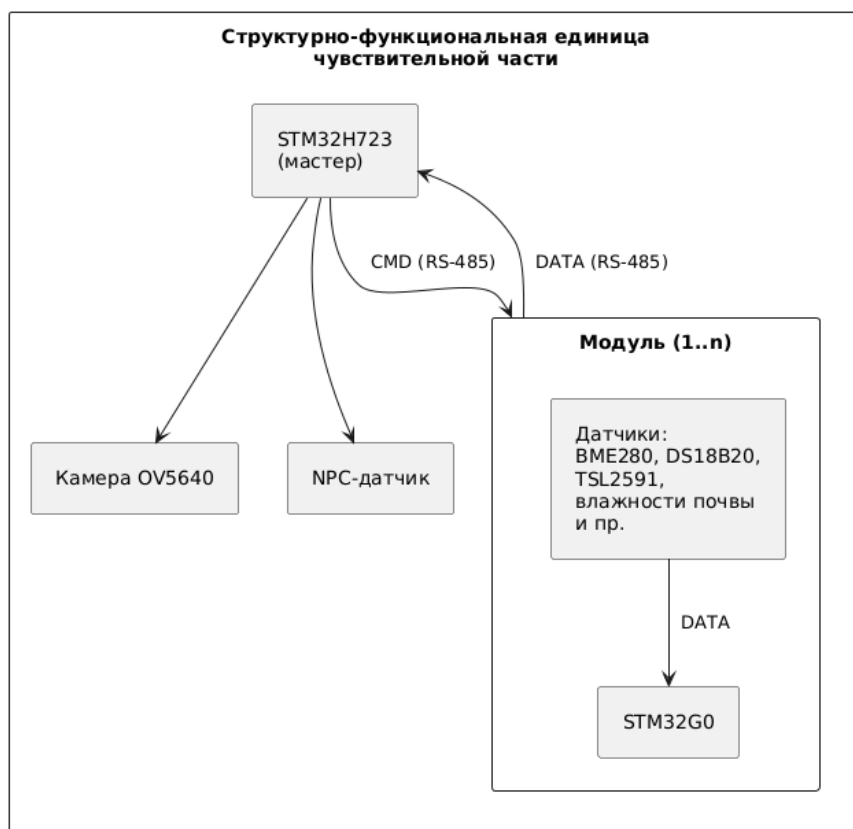


Рисунок 2. - Структурно-функциональная единица чувствительной части

Управляющий модуль (мастер) координирует подчинённые устройства, осуществляет конфигурацию и передаёт данные на сервер (Raspberry Pi 5) через Ethernet-контроллер W5500 с постоянным TCP-соединением. Также он обрабатывает изображения с камеры OV5640 (5 Мп) и данные NPC-сенсора, оценивающего агрохимические параметры почвы (электропроводимость, pH, концентрации калия, фосфора, азота и солей).

Низкоуровневые модули (слейвы) собирают данные с цифровых сенсоров: BME280 (температура, влажность, давление), DS18B20 (температура почвы), TSL2591 (освещённость), ёмкостные датчики (влажность почвы). Обмен между узлами осуществляется по интерфейсу RS-485 с AutoDirection, реализованным на чипе MAX13487EESA+.

Алгоритм работы чувствительной подсистемы включает две фазы: обнаружение (Discovery) и опрос (Polling), реализуемые по интерфейсу RS-485.

Процесс обнаружения и адресации узлов представлен на диаграмме последовательности (рис. 3). После подачи питания мастер-контроллер STM32H723 формирует широковещательный кадр CMD_START_DISCOVERY, задавая параметры тайм-аутов и максимального окна случайной задержки (max_backoff). Каждый слейв STM32G0, получив этот сигнал, активирует режим Listen-Before-Talk с рандомизированным back-off: при свободной шине отправляется сообщение CMD_ANNOUNCE, содержащее уникальный UID и тип сенсора.

Мастер подтверждает получение через CMD_ANNOUNCE_ACK, а по завершении окна сбора формирует пакет CMD_ASSIGN_ADDRS, назначая каждому слейву уникальный адрес. После этого узлы активируют аппаратную фильтрацию UART и переходят в режим ожидания команд.

В фазе опроса (Polling) STM32H7 последовательно рассылает запросы CMD_POLL_DATA на адреса зарегистрированных узлов. Слейв, распознав команду, активирует сенсоры, формирует ответ CMD_DATA_RESPONSE с показаниями и UID, и отправляет его обратно. Мастер обрабатывает ответы и передаёт данные на верхний уровень для анализа. Обе фазы обеспечивают детерминированную регистрацию узлов и энергосберегающий опрос, минимизируя активное время слейвов.

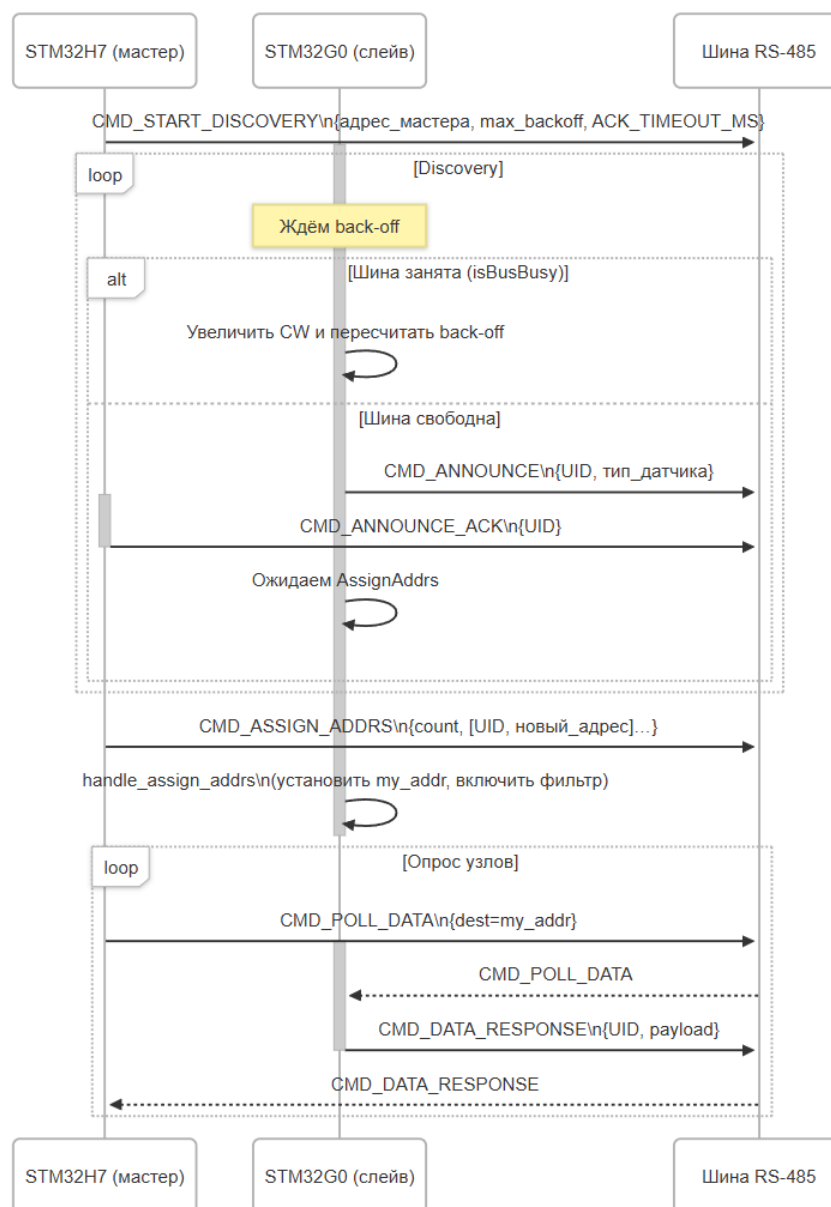


Рисунок 3. - Диаграмме последовательности алгоритма

Анализирующая часть («тело нейрона»). Средний уровень нейроподобной архитектуры («тело нейрона»), реализован на одноплатном компьютере Raspberry Pi 5. Он выполняет функции централизованной обработки, анализа и маршрутизации информации, поступающей от чувствительной части, а также координации работы исполнительных механизмов. Архитектурно этот компонент выступает в роли интеграционного узла между полевыми модулями, нейросетевыми сервисами и пользовательским интерфейсом (рисунок 4).

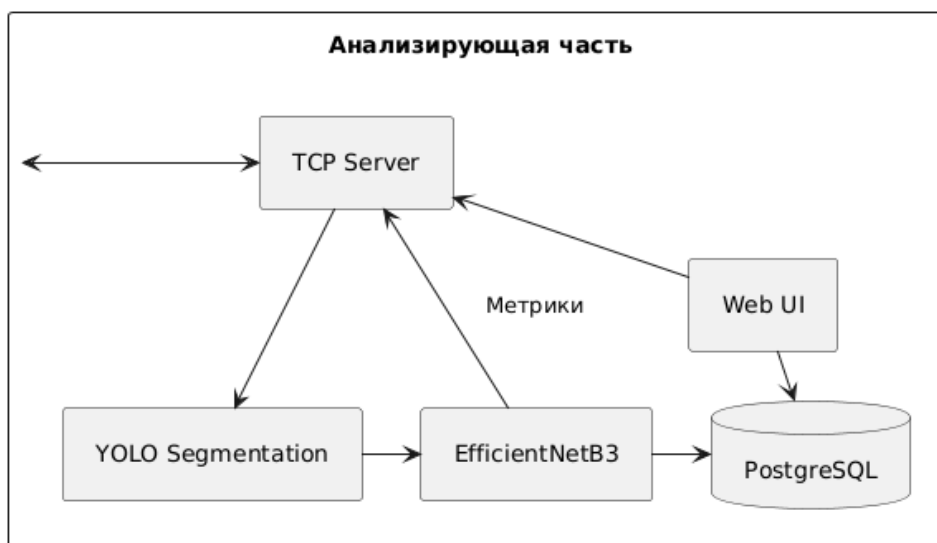


Рисунок 4. — Архитектура анализирующей части («тела нейрона»)

В базовом режиме ядро системы параллельно обрабатывает два потока: телеметрию и изображения, поступающие от модулей через TCP (W5500), и команды пользователя, передаваемые через веб-интерфейс Flask. При поступлении данных активируется режим обработки: изображения декодируются, значения сенсоров нормализуются и сохраняются в базу данных PostgreSQL с привязкой ко времени и источнику.

Затем запускается нейросетевой анализ: YOLO сегментирует листья, а EfficientNetB3 классифицирует признаки заболеваний. Результаты анализа и рекомендации по управлению микроклиматом сохраняются в БД и при необходимости транслируются пользователю в реальном времени.

Подсистема команд принимает HTTP-запросы, сохраняет их в очередь и передаёт C++-сервису через ZeroMQ. После десериализации команда направляется в исполнительный модуль, а результат выполнения фиксируется в БД. По завершении система возвращается в исходное состояние ожидания событий.

Хранилище данных системы построено на основе реляционной модели и реализовано в виде иерархически организованной схемы PostgreSQL, предназначенную для хранения конфигурации устройств, показаний

сенсоров, команд управления и результатов нейросетевого анализа. Центральной сущностью является устройство, объединяющее один или несколько модулей (сенсорных или исполнительных), привязанных к конкретному участку.

База данных содержит информацию о параметрах модулей, показаниях датчиков, действиях исполнительных механизмов, изображениях с камер и результатах их обработки, а также об операциях управления, событиях и уведомлениях.

Архитектура серверного модуля. Серверный модуль реализован в виде четырёхуровневой архитектуры, обеспечивающей высокую производительность и масштабируемость при работе с большим числом устройств. Взаимодействие компонентов показано на рисунке 5.

На нижнем уровне располагаются структуры данных (например, DeviceInfo и пакеты протокола), формализующие сообщения и описывающие подключённые устройства. Хранилище зарегистрированных узлов реализовано через unordered_map, использующую хэш-функции для UID.

Слой парсинга (ProtocolParser) обеспечивает преобразование входящих TCP-пакетов в структуры команд и вызов соответствующих обработчиков (например, on_register, on_data_batch).

Каждое TCP-соединение обрабатывается отдельным объектом TcpSession, в котором реализованы обработчики команд, регистрация устройств, обновление таймингов и формирование ответов. Состояние устройств поддерживается в DeviceRegistry.

Для масштабирования используется пул событийных контекстов (IoContextPool) на базе Boost.Asio, позволяющий равномерно распределять соединения между потоками. Класс TcpServer управляет приёмом подключений и созданием сессий.

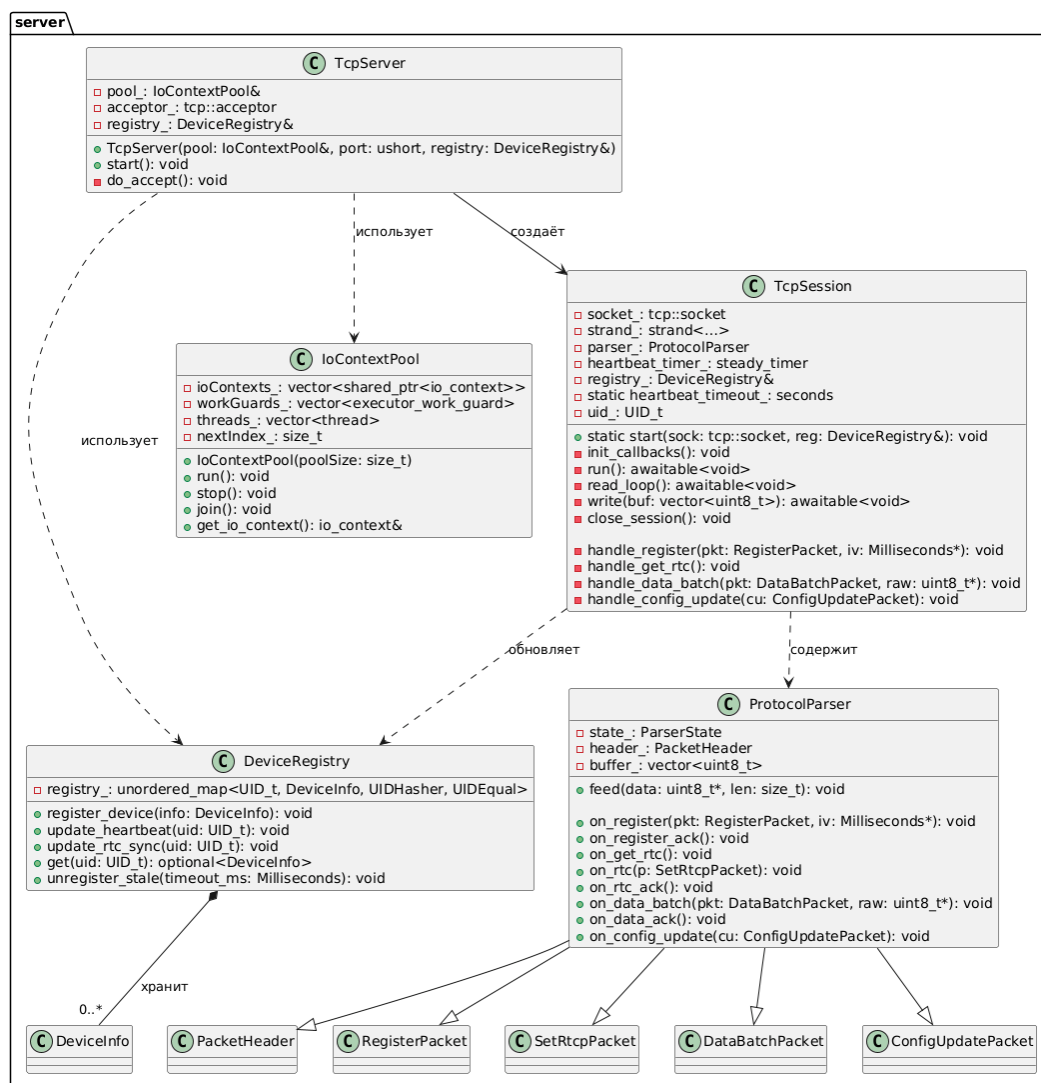


Рисунок 5. — Архитектура серверного модуля и его компонент

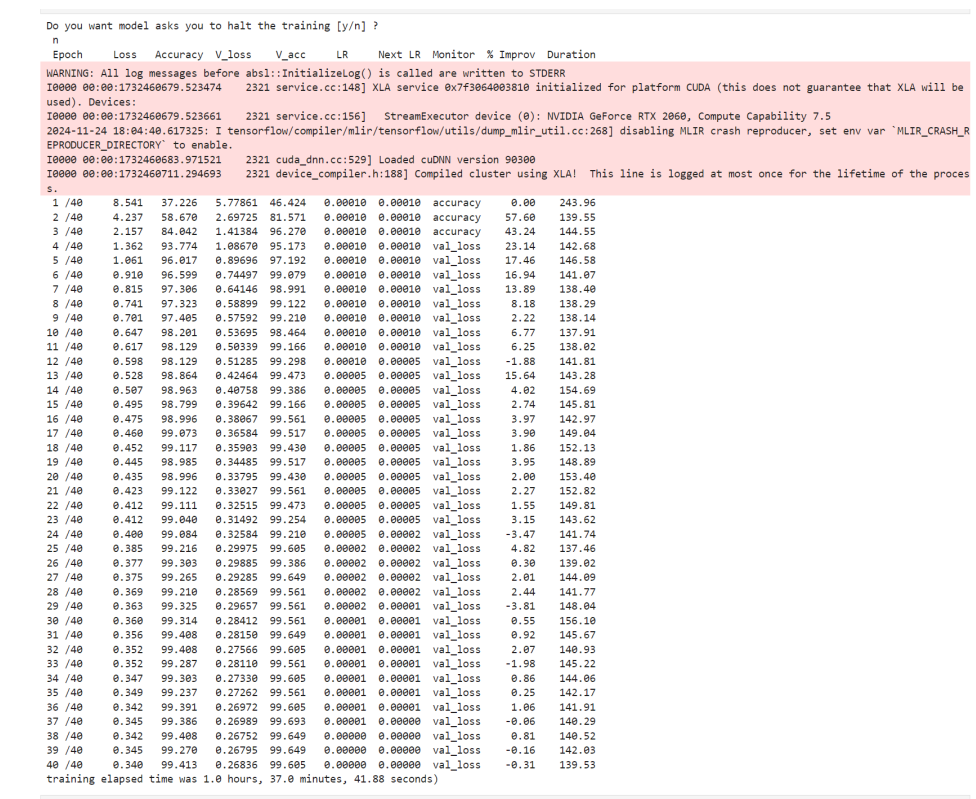
Исполнительный уровень («дендрит нейрона»). Исполнительный уровень реализован в виде распределённой системы модулей, преобразующих аналитические результаты в физические воздействия. На основе метрик нейросетевого анализа формируются команды с заданными параметрами (температура, влажность, освещённость и др.), которые передаются по ТСР на коллекторный контроллер.

Коллектор принимает команды, управляет насосом через преобразователь частоты, маршрутизирует данные и команды к групповым контроллерам и обеспечивает обратную связь с ядром системы. Групповые контроллеры управляют исполнительными устройствами: регулируют краны,

активируют освещение и нагрев, управляют системой подкормки и промывки.

Замкнутый контур управления с датчиками давления и расхода позволяет адаптировать режим работы в реальном времени. Телеметрия также доступна пользователю через интерфейс мониторинга.

Нейросетевой анализ изображения. Эксперименты проводились на рабочей станции с графическим ускорителем NVIDIA GeForce RTX 2060 (Compute Capability 7.5) под управлением TensorFlow 2.18. Обучение модели представлено на рисунке 6.



```

Do you want model asks you to halt the training [y/n] ?
n
Epoch   Loss      Accuracy  V_loss    V_acc     LR        Next LR   Monitor  % Improv  Duration
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1732460679.523474    2321 service.cc:148] XLA service 0x7f3064003810 initialized for platform CUDA (this does not guarantee that XLA will be
used). Devices:
I0000 00:00:1732460679.523661    2321 service.cc:156] StreamExecutor device (0): NVIDIA GeForce RTX 2060, Compute Capability 7.5
2024-11-24 18:04:40.617325: I tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:268] disabling MLIR crash reproducer, set env var 'MLIR_CRASH_R
EPRODUCER_DIRECTORY' to enable.
I0000 00:00:1732460683.971521    2321 cuda_dnn.cc:529] Loaded cuDNN version 90300
I0000 00:00:1732460711.294693    2321 device_compiler.h:188] Compiled cluster using XLA! This line is logged at most once for the lifetime of the proces
s.
1 /40    8.541    37.226    5.77861    46.424    0.00010    0.00010    accuracy    0.00    243.96
2 /40    4.237    58.670    2.69725    81.571    0.00010    0.00010    accuracy    37.60    139.55
3 /40    2.157    84.042    1.41384    95.270    0.00010    0.00010    accuracy    43.24    144.55
4 /40    1.362    93.774    0.80670    95.173    0.00010    0.00010    val_loss    23.14    142.68
5 /40    1.061    96.817    0.80696    97.192    0.00010    0.00010    val_loss    17.46    146.58
6 /40    0.910    96.599    0.74497    99.079    0.00010    0.00010    val_loss    16.94    141.07
7 /40    0.815    97.306    0.64146    98.991    0.00010    0.00010    val_loss    13.89    138.40
8 /40    0.741    97.323    0.58899    99.122    0.00010    0.00010    val_loss    8.18    138.29
9 /40    0.701    97.405    0.57592    99.210    0.00010    0.00010    val_loss    2.22    138.14
10 /40   0.647    98.201    0.53695    98.464    0.00010    0.00010    val_loss    6.77    137.91
11 /40   0.617    98.129    0.50339    99.166    0.00010    0.00010    val_loss    6.25    138.02
12 /40   0.598    98.129    0.51285    99.298    0.00010    0.00005    val_loss    -1.88    141.81
13 /40   0.528    98.864    0.42464    99.473    0.00005    0.00005    val_loss    15.64    143.28
14 /40   0.507    98.963    0.40758    99.386    0.00005    0.00005    val_loss    4.02    154.69
15 /40   0.495    98.799    0.39642    99.166    0.00005    0.00005    val_loss    2.74    145.81
16 /40   0.475    98.996    0.38067    99.561    0.00005    0.00005    val_loss    3.97    142.97
17 /40   0.460    99.073    0.36584    99.517    0.00005    0.00005    val_loss    3.90    149.04
18 /40   0.452    99.117    0.35903    99.430    0.00005    0.00005    val_loss    1.86    152.13
19 /40   0.445    98.985    0.34485    99.517    0.00005    0.00005    val_loss    3.95    148.89
20 /40   0.435    98.996    0.33795    99.430    0.00005    0.00005    val_loss    2.00    153.40
21 /40   0.423    99.122    0.33027    99.561    0.00005    0.00005    val_loss    2.27    152.82
22 /40   0.412    99.111    0.32515    99.473    0.00005    0.00005    val_loss    1.55    149.81
23 /40   0.412    99.040    0.31492    99.254    0.00005    0.00005    val_loss    3.15    143.62
24 /40   0.400    99.084    0.32584    99.210    0.00005    0.00002    val_loss    -3.47    141.74
25 /40   0.385    99.216    0.29975    99.605    0.00002    0.00002    val_loss    4.82    137.46
26 /40   0.377    99.303    0.29885    99.386    0.00002    0.00002    val_loss    0.30    139.02
27 /40   0.375    99.265    0.29285    99.649    0.00002    0.00002    val_loss    2.01    144.09
28 /40   0.369    99.210    0.28569    99.561    0.00002    0.00002    val_loss    2.44    141.77
29 /40   0.363    99.325    0.29657    99.561    0.00002    0.00001    val_loss    -3.81    148.04
30 /40   0.360    99.314    0.28412    99.561    0.00001    0.00001    val_loss    0.55    156.10
31 /40   0.356    99.408    0.28150    99.649    0.00001    0.00001    val_loss    0.92    145.67
32 /40   0.352    99.408    0.27560    99.605    0.00001    0.00001    val_loss    2.07    140.93
33 /40   0.352    99.287    0.28110    99.561    0.00001    0.00001    val_loss    -1.98    145.22
34 /40   0.347    99.303    0.27330    99.605    0.00001    0.00001    val_loss    0.86    144.06
35 /40   0.349    99.237    0.27262    99.561    0.00001    0.00001    val_loss    0.25    142.17
36 /40   0.342    99.391    0.26972    99.605    0.00001    0.00001    val_loss    1.06    141.91
37 /40   0.345    99.386    0.26989    99.693    0.00001    0.00000    val_loss    -0.06    140.29
38 /40   0.342    99.408    0.26752    99.649    0.00000    0.00000    val_loss    0.81    140.52
39 /40   0.345    99.270    0.26795    99.649    0.00000    0.00000    val_loss    -0.16    142.03
40 /40   0.340    99.413    0.26836    99.605    0.00000    0.00000    val_loss    -0.31    139.53
training elapsed time was 1.0 hours, 37.0 minutes, 41.88 seconds)

```

Рисунок 6. – Обучение модели

Для более эффективного исполнения вычислительных графов задействован XLA-компилятор (Accelerated Linear Algebra). В целях воспроизводимости всех детерминированных процедур (инициализация весов, случайное перемешивание батчей, аугментация изображений) был

установлен фиксированный $\text{seed} = 42$ как для Python random, так и для NumPy и TensorFlow.

В качестве основы экстрактора признаков была выбрана сеть EfficientNetB3, предварительно обученная на ImageNet, с отключённым классификационным блоком ($\text{include_top}=\text{False}$). На выходе её последнего сверточного блока формируется тензор размерности 1536. Далее последовательно применялись: BatchNormalization (6 144 параметра) для выравнивания распределения активаций, Dense(256) + ReLU + регуляризация $L2(\lambda=0.016)$ по весам и $L1(\lambda=0.006)$ по смещениям (393 472 параметра) для обучения высокоуровневых признаков, Dropout(0.45) для борьбы с перенастройкой, Dense(15) + softmax (3 855 параметров) для окончательной классификации на 15 классов.

Как видно на рисунке 7, базовый блок EfficientNetB3 насчитывает 10 783 535 параметров, что вместе с дополнительными слоями даёт суммарно 11 187 006 параметров, из которых 11 096 631 обучаемые и 90 375 замороженных.

Для решения задачи многоклассовой классификации использовалась функция потерь categorical_crossentropy и оптимизатор Adam с начальным шагом обучения $\text{learning_rate}=1\times 10^{-4}$, $\beta_1=0.9$, $\beta_2=0.999$. В процессе тренировки были подключены два callback-механизма:

1. EarlyStopping ($\text{monitor}=\text{'val_loss'}$, $\text{patience}=5$, $\text{restore_best_weights}=\text{True}$) — досрочная остановка при отсутствии снижения валидационной потери в течение пяти эпох;

2. ReduceLROnPlateau ($\text{monitor}=\text{'val_loss'}$, $\text{factor}=0.5$, $\text{patience}=3$, $\text{min_lr}=1\times 10^{-6}$) — снижение скорости обучения вдвое при стагнации метрики на валидации.

Перед подачей в модель изображения масштабировались до 224×224 пикселей для унификации входных данных, нормализовались в диапазоне [0,

1] для ускорения сходимости, а также подвергались аугментации «на лету» с использованием `tf.data` — включая горизонтальное отражение с вероятностью 50 % и случайное масштабирование в пределах ± 10 % от исходного размера.

`model.summary()`

Model: "sequential_8"

Layer (type)	Output Shape	Param #
efficientnetb3 (Functional)	(None, 1536)	10,783,535
batch_normalization_9 (BatchNormalization)	(None, 1536)	6,144
dense_16 (Dense)	(None, 256)	393,472
dropout_8 (Dropout)	(None, 256)	0
dense_17 (Dense)	(None, 15)	3,855

Total params: 11,187,006 (42.68 MB)
Trainable params: 11,096,631 (42.33 MB)
Non-trainable params: 90,375 (353.03 KB)

Рисунок 7. - Структура модели: EfficientNetB3 → BatchNorm → Dense(256) → Dropout(0.45) → Dense(15).

На рисунке 8(а) и рисунке 8(б) приведены кривые изменения функции потерь и точности на обучающей и валидационной выборках в ходе 40 эпох. Уже к 5-й эпохе потери резко снижаются: с ≈ 8.5 до 0.8 на обучении и с ≈ 5.5 до 0.7 на валидации, а после 10-й стабилизируются в пределах 0.4–0.3, что свидетельствует о быстром обучении модели. Минимальная валидационная потеря достигается на 38-й эпохе, после чего кривые выравниваются, признаков переобучения не наблюдается. Начальная точность составляет около 40–45 %, но уже к 5-й эпохе обе метрики превышают 95 %, а к 37-й достигается максимум: валидационная точность ≈ 99.6 %. Динамика метрик показывает, что оптимальное состояние достигается в пределах 35–40 эпох, после чего улучшения становятся незначительными и обучение может быть завершено с помощью `EarlyStopping(patience=5)`.

На тестовой выборке 2 279 изображений, по 15 классам и здорового состояния растений, модель показала практически безошибочную классификацию (рисунок 9).

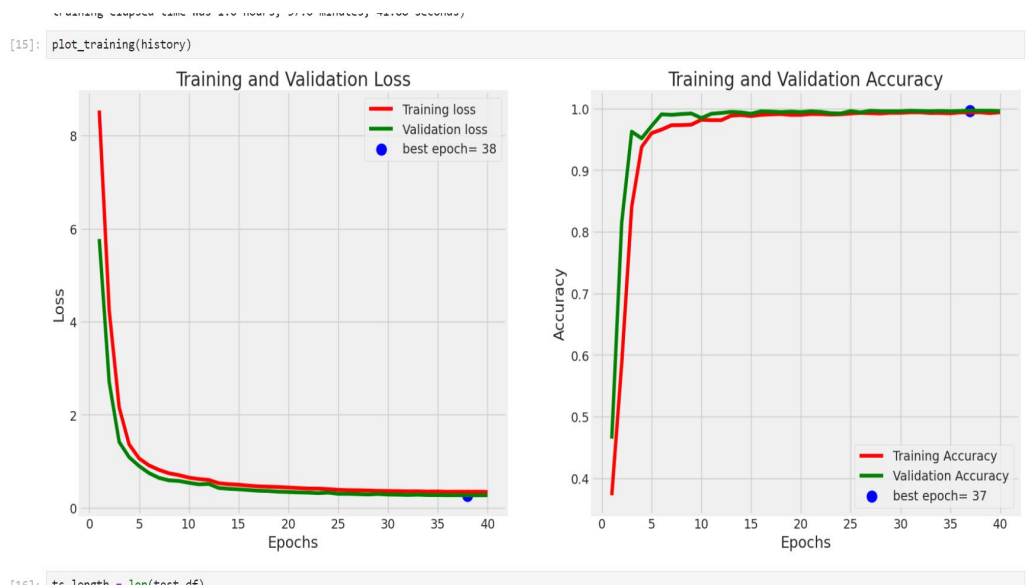


Рисунок 8. - Динамика обучения и валидации нейросети:

а — кривая изменения тренировочной (красная) и валидационной (зелёная) потерь, отметка синим цветом показывает лучшую эпоху;

б — кривая изменения точности на обучающей (красная) и валидационной (зелёная) выборках, отметка синим цветом показывает лучшую эпоху.

Общая точность (ассураку) составила 100 %, при этом значения precision, recall и F1-score для каждого класса находились в диапазоне от 0.99 до 1.00 (рисунок 9). Так, для классов «Pepper__bell__Bacterial_spot», «Pepper__bell__healthy», «Potato__Early_blight» и ряда других precision и recall достигали единицы, а для «Potato__Late_blight», «Tomato__Early_blight» и «Tomato__Target_Spot» значения этих метрик были равны 0.99, что обусловлено округлением при поддержке в 100–191 образец на каждый класс.

Матрица ошибок (рисунок 10) подтверждает согласованность предсказаний — все образцы каждой категории правильно отнесены к своему классу, вне диагонали не обнаружено ни одного ложного срабатывания или пропуска.

```
print(classification_report(test_gen.classes, y_pred, target_names= classes))
```

	precision	recall	f1-score	support
Pepper_bell__Bacterial_spot	1.00	1.00	1.00	99
Pepper_bell__healthy	1.00	1.00	1.00	148
Potato__Early_blight	1.00	1.00	1.00	100
Potato__Late_blight	0.99	1.00	1.00	100
Potato__healthy	1.00	1.00	1.00	15
Tomato__Bacterial_spot	1.00	1.00	1.00	213
Tomato__Early_blight	1.00	1.00	1.00	100
Tomato__Late_blight	1.00	0.99	0.99	191
Tomato__Leaf_Mold	1.00	1.00	1.00	95
Tomato__Septoria_leaf_spot	1.00	1.00	1.00	177
Tomato__Spider_mites Two-spotted_spider_mite	0.99	1.00	1.00	168
Tomato__Target_Spot	1.00	0.99	1.00	141
Tomato__Tomato_Yellow_Leaf_Curl_Virus	1.00	1.00	1.00	536
Tomato__Tomato_mosaic_virus	1.00	1.00	1.00	37
Tomato__healthy	0.99	1.00	1.00	159
accuracy			1.00	2279
macro avg	1.00	1.00	1.00	2279
weighted avg	1.00	1.00	1.00	2279

```
model_name = model_name if hasattr(model, 'name') else 'model'
```

Рисунок 9. – Результаты модели на тестовой выборке

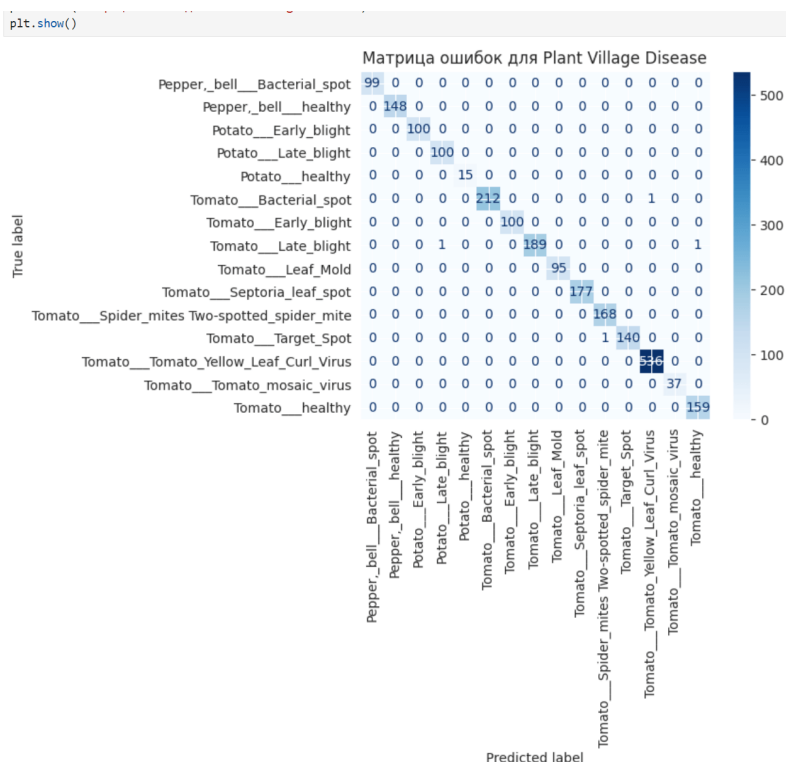


Рисунок 10. – Матрица ошибок для 15 классов

На тестовом стенде среднее время обработки одного изображения (224×224) составило ≈ 68 мс при размере батча = 1. (Рисунок 11а). Это соответствует примерно 15 fps, что позволяет применять модель в режиме реального времени для мониторинга состояния растений.



Рисунок 11. Результаты работы модели и образец поражённого листа:

а — Время инференса,

б — Демонстрация результата.

На рисунке 11(б) приведён скриншот пользовательского интерфейса: после подачи на вход фотографии листа томата система выдаёт «Вид: Томаты» и «Болезнь: Листья, заражённые вирусом жёлтых листьев».

Исследование выполнено за счет гранта Российского научного фонда № 25-21-20073 (<https://rscf.ru/project/25-21-20073/>) и гранта администрации Волгоградской области

Литература

1. United Nations Department of Economic and Social Affairs, Population Division. World Population Prospects 2022: Summary of Results / Statistical Papers – United Nations (Ser. A), Population and Vital Statistics Report. New York: United Nations, August 2022. 52 p. ISBN 978-9210014380. DOI: doi.org/10.18356/9789210014380
2. Nagarsheth S., Agbossou K., Henao N., Bendouma M. The Advancements in Agricultural Greenhouse Technologies: An Energy Management Perspective // Sustainability. 2025. Vol. 17. 3407. DOI: doi.org/10.3390/su17083407.

3. Кокиева Г. Е., Дондоков Ж. Ж. Исследование эффективности управления технологическим объектом теплицы // Международный журнал прикладных наук и технологий «Integral». 2022. № 4. С. 1054–1067. URL: cyberleninka.ru/article/n/issledovanie-effektivnosti-upravleniya-tehnologicheskim-obektom-teplitsy
4. Егунов В. А., Королева И. Ю., Типаев Д. В. Алгоритмы движения мобильного робота с построением карты местности в реальном времени // Инженерный вестник Дона. – 2022. – № 4. URL: ivdon.ru/ru/magazine/archive/n4y2022/7570
5. Буданов А. С., Егунов В. А. Использование углов Эйлера в инерциальных навигационных системах // Инженерный вестник Дона. – 2021. – № 7. URL: ivdon.ru/ru/magazine/archive/n7y2021/7072
6. Chen W.-H., Mattson N. S., You F. Intelligent control and energy optimization in controlled environment agriculture via nonlinear model predictive control of semi-closed greenhouse // Applied Energy. — 2022. — Vol. 320. — Article 119334. — ISSN 0306-2619. — DOI: 10.1016/j.apenergy.2022.119334
7. Platero-Horcajadas M., Pardo-Pina S., Cámara-Zapata J.-M., Brenes-Carranza J.-A., Ferrández-Pastor F.-J. Enhancing Greenhouse Efficiency: Integrating IoT and Reinforcement Learning for Optimized Climate Control Sensors. — 2024. — Vol. 24, no. 24. — Article 8109. doi.org/10.3390/s24248109
8. Jeon Y.-J., Kim J.Y., Hwang K.-S., Cho W.-J., Kim H.-J., Jung D.-H. Machine Learning-Powered Forecasting of Climate Conditions in Smart Greenhouse Containing Netted Melons Agronomy. — 2024. — Vol. 14. — Article 1070. doi.org/10.3390/agronomy14051070
9. Adesanya M. A., Obasekore H., Rabiou A., Na W.-H., Ogunlowo Q. O., Akpenpuun T. D., Kim M.-H., Kim H.-T., Kang B.-Y., Lee H.-W. Deep reinforcement learning for PID parameter tuning in greenhouse HVAC system

energy Optimization: A TRNSYS-Python cosimulation approach Expert Systems with Applications. — 2024. — Vol. 252, Part A.

References

1. United Nations Department of Economic and Social Affairs, Population Division. World Population Prospects 2022: Summary of Results Statistical Papers United Nations (Ser. A), Population and Vital Statistics Report. New York: United Nations, August 2022. 52 p. ISBN 978-9210014380. DOI: doi.org/10.18356/9789210014380.
2. Nagarsheth, S.; Agbossou, K.; Henao, N.; Bendouma, M. Sustainability 2025, 17, 3407. DOI: doi.org/10.3390/su17083407.
3. Kokieva, G.E.; Dondokov, Zh.Zh. Mezhdunarodnyy zhurnal prikladnykh nauk i tekhnologiy “Integral” 2022, no. 4, 1054–1067. URL: cyberleninka.ru/article/n/issledovanie-effektivnosti-upravleniya-tehnologicheskim-obektom-teplitsy
4. Egunov, V.A.; Koroleva, I.Yu., Tipaev, D.V. Inzhenerny vestnik Dona 2022, no. 4. URL: ivdon.ru/ru/magazine/archive/n4y2022/7570
5. Budanov, A.S.; Egunov, V.A. Inzhenerny vestnik Dona 2021, no. 7. URL: ivdon.ru/ru/magazine/archive/n7y2021/7072
6. Chen, W.-H.; Mattson, N.S.; You, F. Applied Energy 2022, 320, 119334. ISSN 0306-2619. DOI: 10.1016/j.apenergy.2022.119334.
7. Platero-Horcajadas, M.; Pardo-Pina, S.; Cámara-Zapata, J.-M.; Brenes-Carranza, J.-A.; Ferrández-Pastor, F.-J. Sensors 2024, 24(24), 8109. DOI: doi.org/10.3390/s24248109.
8. Jeon, Y.-J.; Kim, J.Y.; Hwang, K.-S.; Cho, W.-J.; Kim, H.-J.; Jung, D.-H. Agronomy 2024, 14, 1070. DOI: doi.org/10.3390/agronomy14051070.



9. Adesanya, M.A.; Obasekore, H.; Rabiou, A.; Na, W.-H.; Ogunlowo, Q.O.; Akpenpuun, T.D.; Kim, M.-H.; Kim, H.-T.; Kang, B.-Y.; Lee, H.-W. Expert Systems with Applications 2024, 252, Part A.

Дата поступления: 10.05.2026

Дата публикации: 24.06.2025