

Разработка программно-аппаратного комплекса идентификации строительного мусора с помощью нейронных сетей

А. Н. Ильюшина

*Санкт-Петербургский технический колледж
управления и коммерции*

bdbyu@mail.ru

И. М. Новожилов

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

novozhilovim@list.ru

Аннотация. В статье представлена разработка и реализация системы автоматического распознавания и классификации строительных отходов на основе компьютерного зрения и нейронных сетей. Система предназначена для мониторинга въезжающего и выезжающего транспорта на объектах переработки отходов. В основе решения лежит использование модели YOLO для детекции объектов в реальном времени, обученной на размеченном датасете изображений. Для повышения удобства эксплуатации разработан графический интерфейс на базе библиотеки Tkinter, а также реализована интеграция с мессенджером Telegram для автоматической рассылки уведомлений руководству. Разработка выполнена на языке программирования Python с использованием библиотек OpenCV, Ultralytics и Telebot. Система демонстрирует высокую точность классификации таких категорий отходов, как кирпич, бетон, грунт, дерево, пустой кузов и неопределенный груз.

Ключевые слова: компьютерное зрение, нейронные сети, YOLO, распознавание объектов, строительные отходы, мониторинг отходов, Python, телеграм-бот, графический интерфейс, машинное обучение

I. ВВЕДЕНИЕ

Эффективное управление, учет и переработка строительных отходов представляют собой одну из ключевых задач в контексте устойчивого развития городской инфраструктуры и соблюдения экологических норм. Рост объемов строительства и сноса зданий приводит к образованию значительных масс производных отходов (бетон, кирпич, древесина, грунт), требующих оперативной сортировки, утилизации или вторичного использования. Традиционные методы контроля, основанные на визуальном осмотре персоналом и бумажном документообороте, характеризуются высокой трудоемкостью, субъективностью оценок, рисками ошибок и отсутствием оперативных данных для анализа. Это создает потребность во внедрении автоматизированных систем мониторинга, способных объективно идентифицировать тип отходов в режиме реального времени и интегрировать данные в цифровую экосистему предприятия.

Актуальность разработки новых технических решений в данной области подтверждается активным развитием технологий компьютерного зрения и глубокого обучения. Существующие системы

видеонаблюдения и датчиков, применяемые на пунктах весового контроля, фиксируют лишь факт въезда/выезда и массу груза, но не определяют его состав. В то же время современные нейросетевые модели, такие как семейство YOLO (You Only Look Once), RetinaNet или Faster R-CNN, демонстрируют выдающиеся результаты в задачах детекции и классификации объектов на изображениях и видео. Однако их адаптация для специфической задачи распознавания сыпучих и фрагментированных строительных материалов в условиях меняющегося освещения, ракурсов и наличия помех остается нетривиальной задачей, требующей специальной подготовки данных и тонкой настройки моделей. Существующие коммерческие решения либо являются дорогостоящими, либо не учитывают весь спектр локальных требований, таких как интеграция с популярными мессенджерами для оповещения.

Целью данной работы является разработка и практическая реализация программно-аппаратного комплекса для автоматического распознавания, классификации строительных отходов и оперативного информирования ответственных лиц на основе технологий компьютерного зрения и искусственного интеллекта.

Для достижения поставленной цели в исследовании решаются следующие задачи:

1. Провести анализ существующих технических и программных решений в области мониторинга отходов и технологий компьютерного зрения, обосновав выбор архитектуры нейронной сети и стека технологий для разработки.

2. Сформировать репрезентативный и размеченный датасет изображений строительных отходов (категории: «пустой кузов», «кирпич», «бетон», «грунт», «дерево», «неопределенный груз»), адаптированный для обучения детекционной модели.

3. Обучить и протестировать модель нейронной сети на основе фреймворка YOLO для точной детекции и классификации типов отходов в видеопотоке с камеры наблюдения.

4. Разработать кроссплатформенное графическое интерфейсное приложение на Python с использованием библиотеки Tkinter, обеспечивающее удобное управление системой, отображение видеопотока и результатов распознавания.

5. Реализовать модуль автоматической отправки структурированных уведомлений в мессенджер Telegram, содержащих информацию о времени, типе отходов и видеодоказательствах.

6. Провести комплексное тестирование и демонстрацию работоспособности системы в условиях, имитирующих реальную эксплуатацию на пункте контроля, оценив точность классификации и надежность работы всех компонентов.

Научная новизна работы заключается в комплексном подходе, объединяющем адаптацию современной детекционной модели YOLO для задачи распознавания нестандартных объектов (строительные отходы), разработку специализированного графического интерфейса для оператора и создание механики автоматических оповещений через популярный коммуникационный канал, что в совокупности формирует готовое к внедрению интеллектуальное решение для урбанистической экосистемы.

Практическая значимость исследования заключается в создании прототипа системы, который может быть развернут на реальных объектах по приему и переработке строительного мусора. Внедрение системы позволит минимизировать человеческий фактор, ускорить процесс учета, повысить прозрачность операций и обеспечить руководство актуальными данными для принятия управленческих решений, способствуя оптимизации логистики и повышению экологической ответственности предприятия.

II. ВЫБОР СБОРКИ ЯЗЫКА ПРОГРАММИРОВАНИЯ

Современная разработка программного обеспечения предлагает широкий спектр языков программирования — от низкоуровневых C++ и Rust до высокоуровневых Java, C# и JavaScript — для реализации системы распознавания строительных отходов с использованием нейронных сетей и компьютерного зрения оптимальным выбором является именно Python. Этот выбор обусловлен не столько отсутствием альтернатив, сколько уникальным сочетанием характеристик, которые делают Python наиболее подходящим инструментом для решения исследовательских и прикладных задач в области искусственного интеллекта.

В отличие от компилируемых языков, где строгая типизация и необходимость явного управления памятью могут замедлять итерационный процесс, Python предлагает динамическую типизацию и интерпретируемость, что значительно ускоряет прототипирование, эксперименты с архитектурой нейросети и настройку гиперпараметров. В то время как на C++ или Rust потребовалось бы существенно больше времени для реализации и отладки аналогичного функционала, Python позволяет быстро интегрировать готовые модули и сосредоточиться на логике приложения, а не на технических деталях реализации алгоритмов. Кроме того, экосистема Python в области машинного обучения и анализа данных не имеет равных. Языки вроде Java или C#, безусловно, мощны и широко используются в enterprise-разработке, однако их библиотеки для компьютерного зрения (например, OpenCV для Java) и глубокого обучения (как DeepLearning4J) не столь развиты, активно обновляемы и популярны в научном сообществе, как их Python-

аналоги. То же самое касается специализированных сред, таких как MATLAB или R: они отлично подходят для математического моделирования и статистики, но их возможности по созданию полноценных приложений с графическим интерфейсом, сетевыми уведомлениями и работой с видеопотоком в реальном времени — ограничены по сравнению с Python.

Ещё одним ключевым преимуществом является уровень поддержки и интеграции. Большинство современных исследовательских работ, открытых моделей (включая YOLO) и обучающих материалов публикуются именно с примерами на Python. Облачные платформы для машинного обучения, такие как Google Colab, изначально заточены под Python, что упрощает процесс обучения моделей без необходимости настройки локального окружения. В то же время, Python сохраняет баланс между простотой и производительностью: критичные по скорости операции (обработка кадров в OpenCV, инференс нейросети) выполняются на оптимизированных низкоуровневых библиотеках, написанных на C/C++, что позволяет достичь высокой эффективности без потери удобства разработки.

Таким образом, хотя технически данную задачу можно было бы решить и на других языках, выбор Python представляется наиболее прагматичным и эффективным. Он минимизирует накладные расходы на разработку, предоставляет доступ к богатейшему набору инструментов и библиотек, обеспечивает тесную интеграцию со всеми необходимыми компонентами системы — от обучения модели до рассылки уведомлений — и позволяет создавать работающий прототип в сжатые сроки. Именно поэтому Python, несмотря на существование множества альтернатив, остаётся безусловным лидером и наиболее обоснованным выбором для проектов на стыке компьютерного зрения, машинного обучения и прикладной автоматизации.

III. РАЗРАБОТКА ПРОГРАММНОГО КОДА

Разработка кода проходила на базе интегрированной среды разработки PyCharm.

Код на Python представляет собой реализацию системы распознавания строительных отходов с использованием камеры Razer Kiyo. Вот объяснение основных компонентов кода:

1. Импорт библиотек:

- OS - для работы с операционной системой (например, чтение файлов и путей к ним).
- CV2 - библиотека OpenCV для обработки изображений и видео.
- Tkinter-библиотека для создания графического интерфейса, для удобства работы с программой.
- Numpy - библиотека для работы с многомерными массивами и матрицами.
- CVzone - библиотека для работы с изображениями и графическими элементами.
- Ultralytics-библиотека для работы с моделью нейросети.
- Telebot-библиотека для работы с telegram-ботом.

- Часть программного кода данного компонента прописана в приложении А.
- Связывание программы с токеном telegram-бота.
- Получение токена в telegram-боте «BotFather».
- Вписывание токена в программу.

Часть программного кода данного компонента прописана в приложении Б.

2. Создание удобного графического интерфейса для работы с программой:

- Создание видеоплеера, с возможностью взаимодействия с видео, внутри графического интерфейса.
- Создание зоны для загрузки картинок внутри графического интерфейса.
- Красивое оформление визуальной составляющей графического интерфейса.
- Создание возможности взаимодействия с камеры видеонаблюдения и графического интерфейса.

Часть программного кода данного компонента прописана в приложении В.

3. Инициализация камеры:

- Создается объект сар для захвата видео с камеры.
- Устанавливаются параметры ширины и высоты видео.

Часть программного кода данного компонента прописана в приложении Г.

4. Основной процесс обработки видео:

- В цикле происходит захват видео с камеры и его обработка.
- При помощи библиотеки opencv, происходит видоизменение рамки, которая обводит обнаруженный объект.

Теперь поэтапно следуют описать шаги разработки по плану проекта. Разработка программы началась с того, что был найден «датасет» с большим количеством видео. На видео был наложен эффект «Рыбьего глаза» для того, чтобы сузить зону работы нейронной сети для меньшей нагрузки на систему пример показан на рис. 1.



Рис. 1. Пример кадра из видео с наложением эффекта «Рыбий глаз»

После этого было принято решение рассортировать видео на 6 категорий:

- Пустой кузов.
- Кирпич.
- Бетон.
- Грунт.
- Дерево.

Неопределенный груз (тот случай, когда даже человеку не понять, какая категория находится в кузове).

Следующим этапом был процесс разрезания всех видео покадрово для обучения нейронной сети на фотографиях. Были выбраны те кадры, на которых грузовик въезжает четко в середину кадра, это будет использоваться для более точного распознавания категории отходов. Пример кадра показан на рис. 2.

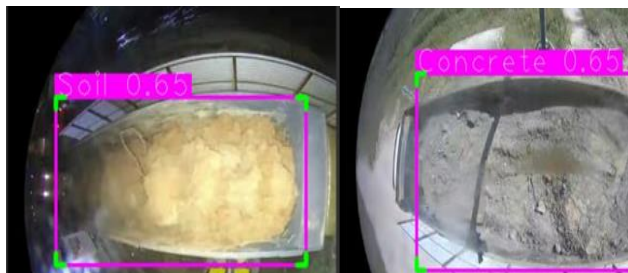


Рис. 2. Пример кадров картинок из видео с определением грунта и бетона

После этого была необходима разметка объектов на фотографии. Для этого использовался онлайн-сервис Roboflow. Roboflow — это платформа для разработчиков компьютерного зрения, позволяющая улучшить сбор данных, их предварительную обработку и методы обучения моделей.

Разметка объектов нужна для того, чтобы указать нейронной сети на то, какой объект ей нужно научиться определять пример платформы показан на рис. 3.

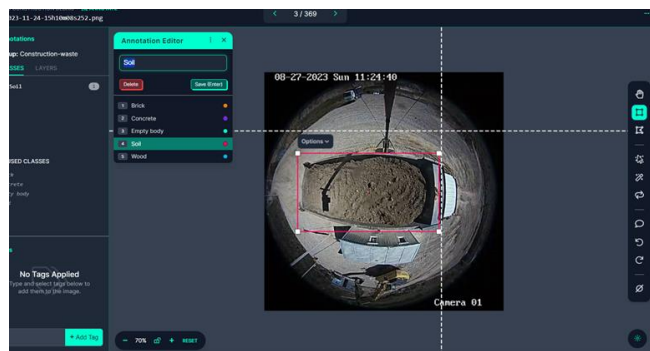


Рис. 3. Процесс разметки объектов

Нейронную сеть необходимо было научить классифицировать определять класс объектов на изображениях. Была проведена работа, примерно над 3000 фотографий, проведена разметка каждой фотографии. Каждой разметке объектов был присвоен свой класс, определенный ранее, это пустой кузов, кирпич, бетон, грунт, дерево, неопределенный груз.

После того, как разметка была окончена, на платформе Roboflow формируется готовый «датасет» для обучения нейронной сети. Датасет представляет

собой массивы, которые были сделаны из фотографий. Массивы нужны для того, чтобы на входе в нейронную сеть она могла их прочитать.

Обучение нейронной сети потребляет достаточно много мощности. Поэтому обучение было проведено в онлайн редакторе кода «Google Collab». Google Colab — это бесплатная среда для разработки и выполнения программного кода в облаке пример среды показан на рисунке 4

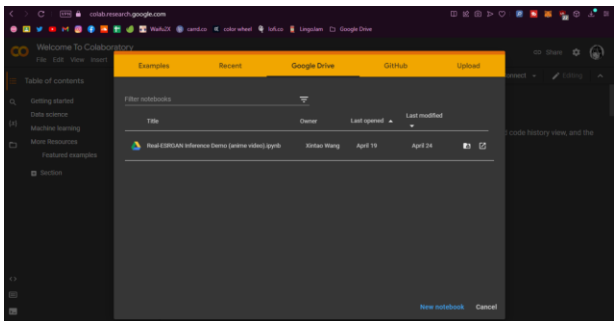


Рис. 4. Пример среды Google Collab

Она предоставляет возможность писать и запускать код на языке Python, используя только браузер, без установки специальных программ на компьютер.

Такое решение было принято, так как вычислительные процессы выполнялись на серверах компании Google, позволяя быстрее обучить нейронную сеть. Примерный фрагмент программного кода обучения нейронной сети прописан в приложении Е.

Далее следовало загрузить готовый «датасет» в Google Disk для того, чтобы код из Google Collab мог с ним взаимодействовать. В коде использовался язык программирования Python. И элемент YOLO из библиотеки ultralytics. YOLO – это система (сеть) обнаружения объектов. Она была разработана Джозефом Редмоном в 2016 году. Наибольшим преимуществом YOLO над другими архитектурами является скорость. Модели семейства YOLO исключительно быстры и намного превосходят R-CNN (Region-Based Convolutional Neural Network) и другие модели.

Это позволяет добиться обнаружения объектов в режиме реального времени. Выделение объектов при помощи YOLO показано на рис. 5.



Рис. 5. Выделение объектов при помощи YOLO

Было принято решение провести 50 эпох обучения нейросети, это означало, что она обучалась 50 раз на одних и тех же картинках. Процесс обучения показан на рис. 6.

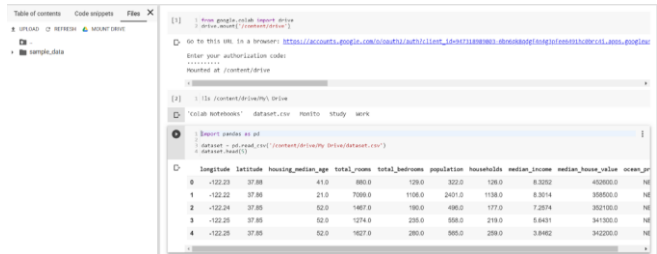


Рис. 6. Процесс обучения нейронной сети

После того, как обучение было завершено, Google Collab дал готовую модель нейронной сети. Эта модель будет использоваться в коде для данного проекта.

Для того чтобы пользователю было удобно взаимодействовать с программой, было принято решение сделать графический интерфейс. Для этого использовалась библиотека Tkinter. Tkinter – это кроссплатформенная библиотека для разработки графического интерфейса на языке Python (начиная с Python 3.0 переименована в Tkinter). Tkinter расшифровывается как Tk interface, и является интерфейсом к Tcl/Tk. Tkinter входит в стандартный дистрибутив Python. Написанный графический интерфейс на языке Python представлен на рис. 7.

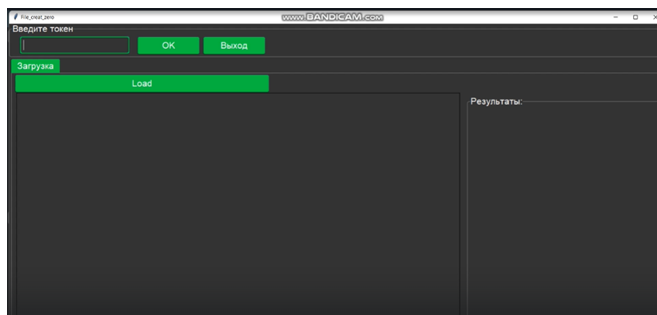


Рис. 7. Графический интерфейс проекта

В графический интерфейс была встроена возможность для включения видео, изображений и трансляции видеопотока с камер видеонаблюдения.

После включения трансляции или загрузки видео, нейросеть начинала свою работу и определяла, и классифицировала объект на выбранном пользователем файле.

Далее была добавлена библиотека openCV, которая визуальна могла улучшить выделение объекта на фото или видео. Выделение объекта при помощи YOLO и openCV показано на рис. 8.



Рис. 8. Выделение объекта при помощи YOLO и OpenCV

Для более оперативной работы в строительной компании было принято решение встроить автоматическую рассылку уведомлений для сотрудников (руководству). Она должна быть реализована в современной социальной сети, которой всем будет удобно пользоваться. Для данных задач идеально подходил мессенджер telegram, так как там была возможность создать своего бота, который может выполнять ту или иную функцию. Для того, чтобы создать данного бота, необходимо получить токен от главного бота в мессенджер telegram, который называется BotFather.

Получения токена в мессенджер telegram показан на рис. 9.

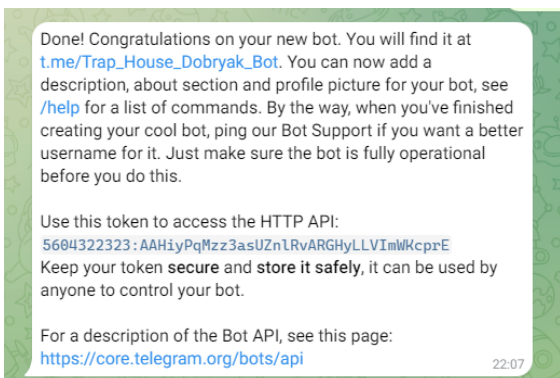


Рис. 9. Получение токена в мессенджер telegram

После этого нужно было вписать токен в программу для взаимодействия функций между собой. Написание токена в коде показано на рис. 10.

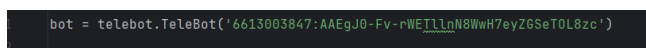


Рис. 10. Написание токена в коде

Далее было необходимо запрограммировать бота на отсылку уведомлений руководству строительных

компаний. Далее необходимо перейти к тестированию и демонстрации проекта.

IV. ДЕМОНСТРАЦИЯ РАБОТЫ

После запуска токена графический интерфейс определяет по видео класс объекта ввозимого/вывозимого строительного мусора. Пример показан на рис. 11

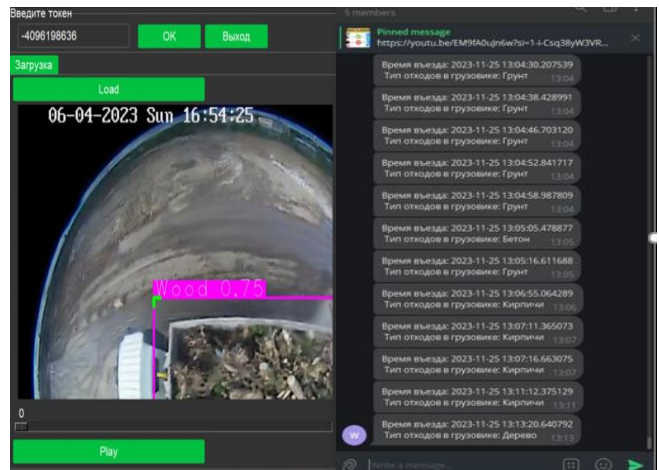


Рис. 11. Определение по видео класс строительного отхода «Дерево»

На рис. 12 показан класс объекта ввозимого/вывозимого строительного мусора «Бетон».

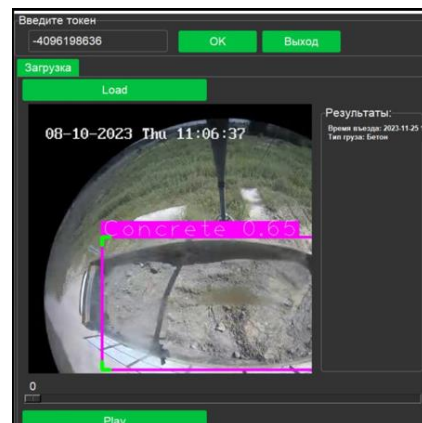


Рис. 12. Определение по видео класс строительного отхода «Бетон»

На рис. 13 показан класс объекта ввозимого/вывозимого строительного мусора «Грунт».

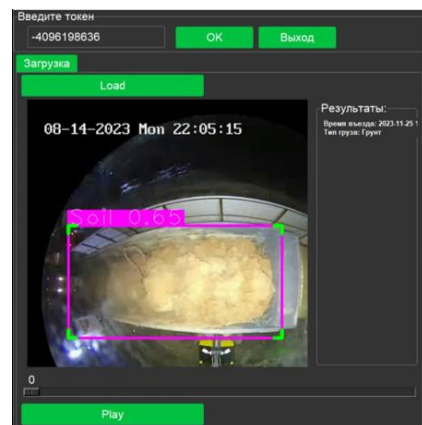


Рис. 13. Определение по видео класс строительного отхода «Грунт»

На рис. 14 показан класс объекта ввозимого/вывозимого строительного мусора «Кирпич».

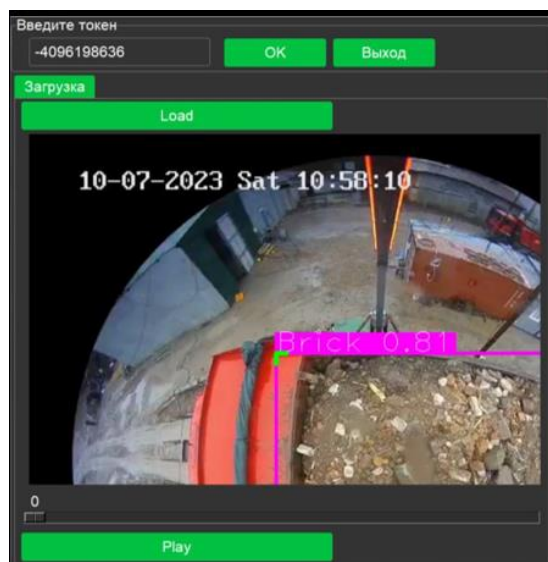


Рис. 14. Определение по видео класс строительного отхода «Кирпич»

В ходе демонстрации результатов проекта система показала хорошие результаты по захвату видео с камеры, система четко определяла тип ввозимого/вывозимого строительного мусора, взаимодействуя с мессенджер telegram, руководству (сотрудникам), приходило уведомление с датой, временем, типом строительного мусора ввозимого или вывозимого в зону переработки.

V. ЗАКЛЮЧЕНИЕ

В ходе исследования была успешно разработана и протестирована система автоматического распознавания и классификации строительных отходов. Система интегрирует в себе современные технологии компьютерного зрения (YOLO, OpenCV), машинного обучения и средства автоматизации коммуникаций (Telegram Bot API). Реализованное решение позволяет в реальном времени детектировать и классифицировать тип мусора в кузове грузового автомобиля с последующей отправкой структурированного уведомления ответственному персоналу. Практические испытания подтвердили работоспособность системы и её точность в определении заданных категорий отходов. Научная новизна работы заключается в адаптации и применении архитектуры YOLO для задачи классификации строительного мусора в условиях урбанистической среды. Перспективы развития системы видятся в создании централизованной базы данных для анализа потока отходов, а также в использовании беспилотных летательных аппаратов для расширения зоны видеомониторинга. Разработанный комплекс представляет собой готовое к внедрению решение, способное повысить эффективность контроля за обращением с отходами на строительных и перерабатывающих объектах.

СПИСОК ЛИТЕРАТУРЫ

[1] Marinina O.A., Ilyushin Y.V., Kildyushov E.V. (2025). Comprehensive Analysis and Forecasting of Indicators of Sustainable Development of Nuclear Industry Enterprises. *International Journal of Engineering Transactions B Applications*, 38(11), 2527-2536. <https://doi.org/10.5829/ije.2025.38.11b.05>

[2] Ilyushin Y.V., Novozhilov I.M. (2020). Temperature Field Control of a Metal Oil-well Tubing for Producing of High-Paraffin Oil. *Proceedings of 2020 23rd International Conference on Soft Computing and Measurements Scm 2020*, 149-152. <https://doi.org/10.1109/SCM50615.2020.9198816>

[3] Ilyushin Y., Novozhilov I.M. (2019). Automation of the Paraffin Oil Production Technological Process. *Proceedings of 2019 3rd International Conference on Control in Technical Systems Cts 2019*, 164-167. <https://doi.org/10.1109/CTS48763.2019.8973352>

[4] Ilyushin Y.V., Novozhilov I.M. (2017). Software implementation of a pulse regulator of a distributed distributed control object. *Proceedings of 2017 IEEE 2nd International Conference on Control in Technical Systems Cts 2017*, 315-317. <https://doi.org/10.1109/CTS2017.8109555>

[5] Ilyushin Y.V., Afanaseva O.V. (2019). Development of the drilling screws temperature field pulse management for the Kislovodsky narzan deposit hot mineral water mining. *International Multidisciplinary Scientific Geoconference Surveying Geology and Mining Ecology Management Sgem*, 19(2.1), 849-856.

[6] Ilyushin Y.V., Novozhilov I.M. (2019). Methodology of inspection of absolute stability of pulse distributed control system. *Proceedings of 2019 22nd International Conference on Soft Computing and Measurements Scm 2019*, 102-106. <https://doi.org/10.1109/SCM.2019.8903839>

[7] Ilyushin Y.V., Pervukhin D.A., Afanaseva O.V., Afanasyev M.P., Kolesnichenko S.V. (2015). The methods of the synthesis of the nonlinear regulators for the distributed one-dimension control objects. *Journal of Theoretical and Applied Information Technology*, 82(1), 54-67.

[8] Ilyushin Y.V., Mokeev A. (2019). Distribution of temperature in a spatially onedimensional object as a result of the active point source. *Arpn Journal of Engineering and Applied Sciences*, 14(6), 1238-1243.

[9] Ilyushin Y., Mokeev A. (2017). Tunnel furnace of a conveyor type: Technical controlling of the temperature field. *International Journal of Applied Engineering Research*, 12(20), 9377-9389.

[10] Ilyushin Y.V., Afanaseva O.V. (2019). Multithreading analysis of seismic data on the hybrid supercomputer. *International Multidisciplinary Scientific Geoconference Surveying Geology and Mining Ecology Management Sgem*, 19(2.1), 973-978.

[11] Ilyushin Y.V., Novozhilov I.M. (2017). Analyzing of distributed control system with pulse control. *Proceedings of 2017 20th IEEE International Conference on Soft Computing and Measurements Scm 2017*, 296-298. <https://doi.org/10.1109/SCM.2017.7970565>

[12] Ilyushin Y.V., Novozhilov I.M. (2017). Analyzing of heating elements' location of distributed control objects. *Proceedings of 2017 20th IEEE International Conference on Soft Computing and Measurements Scm 2017*, 138-141. <https://doi.org/10.1109/SCM.2017.7970519>

[13] Ilyushin Y.V., Mokeev A. (2017). The control system of the thermal field in tunnel furnace of a Conveyor type. *Arpn Journal of Engineering and Applied Sciences*, 12(22), 6595-6605.

[14] Ereemeeva A.M.; Ilyushin Y.V. Temperature Control During Storage of Raw Materials in the Process of Biodiesel Fuel Production. *Inventions* 2025, 10, 7. <https://doi.org/10.3390/inventions10010007>

[15] Martirosyan A., Ilyushin Y., Afanaseva O., Kukharova T., Asadulagi M., & Khloponina V. (2025). Development of an Oil Field's Conceptual Model. *International Journal of Engineering*, 38(2), 381-388. <https://doi.org/10.5829/ije.2025.38.02b.12>

[16] Kukharova T., Martirosyan A., Asadulagi M.-A., Ilyushin Y. Development of the Separation Column's Temperature Field Monitoring System. *Energies* 2024, 17, 5175. <https://doi.org/10.3390/en17205175>

[17] Kukharova T., Maltsev P., Novozhilov I. Development of a Control System for Pressure Distribution During Gas Production in a Structurally Complex Field. *Appl. Syst. Innov.* 2025, 8, 51. <https://doi.org/10.3390/asi8020051>

[18] Kleshnia V.A., Kukharova T., Fedosov I.S., Tsapleva V.V. (2025). Modeling of Pressure Control System in Oil Wells Accounting for Reservoir Non-Homogeneity, 64-67. <https://doi.org/10.1109/CTS67336.2025.11196275>

[19] Kukharova T.V., Utkin V.A., Boev I.V. Observation and Prediction Systems Modeling for Human Mental State. *2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018*, 2018, 8602831. <https://doi.org/10.1109/FarEastCon.2018.8602831>

[20] Ilyushin Y.V., Boronko E.A. Development of a Mathematical Model of the Electromagnetic Field Formation Process Based on System Analysis Methods. *Mathematics* 2026, 14, 399. <https://doi.org/10.3390/math14030399>