

Интеллектуальный контроль производственного брака деталей на основе YOLO: автоматическое выявление сколов, трещин и нарушений геометрии

Саухин Сергей Иванович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассмотрена система визуального контроля производственного брака деталей на основе YOLO. Описаны сбор и разметка изображений, обучение модели YOLOv8n, локализация дефектов, запись событий в SQLite и отображение статистики в веб-дашборде. Работа ориентирована на пилотное внедрение контроля сколов, трещин и нарушений геометрии.

Ключевые слова: компьютерное зрение, промышленный контроль качества, YOLO, дефекты деталей, машинное зрение, нейронные сети, автоматизация производства.

YOLO-based intelligent visual inspection system for production defects: detection of chips, cracks and geometry violations in parts

Saukhin Sergey Ivanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The paper describes a YOLO-based visual inspection system for detecting production defects in parts. The implementation includes image annotation, YOLOv8n training, defect localization, SQLite logging and dashboard visualization. The system is aimed at pilot inspection of chips, cracks and geometry violations.

Keywords: computer vision, industrial quality control, YOLO, part defects, machine vision, neural networks, production automation.

1 Введение

1.1 Актуальность

Контроль качества является одной из ключевых операций производственного процесса. Даже небольшие дефекты деталей могут привести к нарушению сборки, снижению надежности изделия, возвратам продукции и дополнительным затратам на исправление брака. Особенно критичны дефекты, которые визуально заметны, но могут быть пропущены при поточном ручном контроле: сколы кромок, трещины, повреждения поверхности и отсутствие технологических элементов.

На практике многие предприятия сталкиваются с противоречием: промышленная система машинного зрения обеспечивает высокую надежность контроля, но требует значительных затрат на оборудование, настройку, интеграцию и сопровождение. Для малых производственных участков или опытных партий такие затраты часто оказываются неоправданными. При этом полностью ручной контроль не позволяет стабильно поддерживать одинаковое качество проверки на протяжении всей смены.

Развитие нейросетевых методов компьютерного зрения позволяет создать промежуточное решение: доступный программно-аппаратный комплекс, который не заменяет промышленную линию целиком, но автоматизирует обнаружение наиболее типичных дефектов. Такой подход особенно полезен на этапе пилотного внедрения, когда необходимо быстро проверить применимость компьютерного зрения к конкретной детали и оценить экономический эффект от автоматизации контроля [6, 7].

Алгоритмы семейства YOLO позволяют решать задачу детекции объектов в режиме, близком к реальному времени. В отличие от простой классификации изображения, модель детекции возвращает координаты дефекта, его класс и уверенность распознавания. Это делает результат понятным для оператора: система не только сообщает о браке, но и показывает конкретную проблемную область [1, 8].

1.2 Производственная проблема

В данной работе рассматривается типовая производственная ситуация: на участке выпускаются небольшие пластиковые детали, качество которых проверяется визуально. Основные дефекты связаны с механическим повреждением края, появлением трещин и нарушением геометрии, например отсутствием отверстия. Такие дефекты могут возникать из-за износа инструмента, неправильной настройки оборудования, ошибок транспортировки или нестабильности технологического процесса.

Проблема состоит не только в самом факте брака, но и в отсутствии оперативной статистики. При ручной проверке оператор обычно отбраковывает изделие, но данные о типе дефекта, времени возникновения и частоте повторения фиксируются нерегулярно. В результате предприятие теряет возможность быстро увидеть, какой дефект становится массовым и на каком этапе необходимо искать причину.

Предлагаемая система направлена на решение этой практической проблемы. Она должна автоматически обнаруживать дефекты, записывать результат проверки в журнал и формировать статистику по типам брака. Таким образом, система выполняет сразу две функции: помогает оператору при контроле конкретной детали и создает цифровой след для последующего анализа качества производственного процесса.

Опора на YOLO в данной задаче не является произвольным выбором. В российской статье М.А. Астаповой и М.Ю. Уздяева, опубликованной в журнале «Моделирование, оптимизация и информационные технологии» и доступной через eLIBRARY/РИНЦ, нейронные сети семейства YOLO

применялись для детектирования дефектов неисправных элементов линий электропередач. Авторы сравнивали YOLOv3, YOLOv4 и их tiny-версии; по результатам эксперимента YOLOv3 показала precision $0,96 \pm 0,03$, recall $0,95 \pm 0,02$, mAP@0,50 $0,97 \pm 0,02$ и mAP@0,75 $0,78 \pm 0,03$ [1]. Эти данные показывают, что YOLO уже применяется в задачах технического зрения, где требуется находить дефектные области на изображении, а не просто классифицировать объект целиком.

1.3 Цель, задачи и практическая значимость

Целью исследования является разработка и оценка интеллектуальной системы визуального контроля производственного брака деталей на основе YOLO, обеспечивающей локализацию дефектов, регистрацию событий и визуальное представление статистики брака.

Для достижения цели были поставлены следующие задачи:

- сформулировать требования к системе визуального контроля для производственного участка;
- подготовить датасет изображений исправных и дефектных деталей;
- разметить дефекты в формате YOLO с учетом локализации поврежденных областей;
- обучить модель YOLOv8n и оценить качество детекции;
- реализовать модуль регистрации дефектов в базе данных;
- разработать веб-дашборд для отображения статистики контроля качества;
- проанализировать ограничения модели и возможности практического внедрения.

Научно-практическая значимость работы заключается в построении воспроизводимой методики быстрого развертывания нейросетевого визуального контроля для конкретной детали и ограниченного набора дефектов. Предложенный подход может использоваться как пилотная модель перед внедрением более сложных промышленных решений, что согласуется с исследованиями по применению искусственного интеллекта на российских производственных предприятиях [6, 7].

2 Материалы и методы

2.0 Фактическая база метода

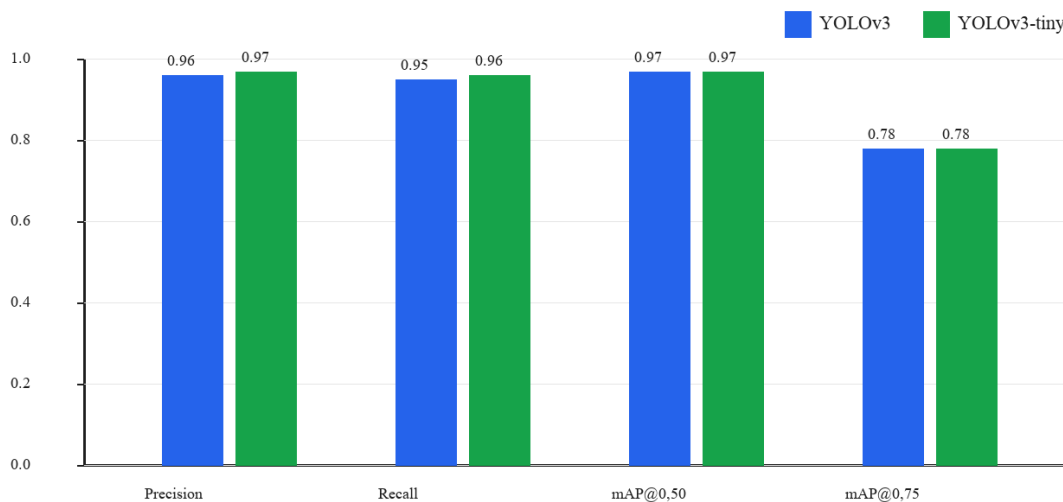
Фактическая база выбранного метода строится на трех положениях. Во-первых, сверточные нейронные сети применяются для детекции визуальных объектов и дефектных областей, что отмечается в российских публикациях по техническому зрению и неразрушающему контролю [1, 2, 4, 5]. Во-вторых, семейство YOLO относится к одностадийным детекторам: модель за один проход по изображению формирует рамки, классы и значения уверенности [8]. В-третьих, качество таких моделей оценивается не субъективно, а по воспроизводимым метрикам: precision, recall, mAP@0,50 и mAP@0,75 или mAP@0,50:0,95 [1, 8].

Работа М.А. Астаповой и М.Ю. Уздяева важна для данной статьи как российский пример применения YOLO именно к задаче обнаружения дефектов. В ней подчеркивается необходимость выбора метода, который сочетает скорость и точность обнаружения объектов. Для YOLOv3 и YOLOv3-tiny в задаче дефектов ЛЭП были получены близкие высокие значения метрик, а авторы сделали вывод о перспективности более простой сети ввиду меньшего размера и более высокой скорости обработки информации [1]. Этот вывод напрямую связан с предлагаемым прототипом: для пилотного производственного контроля рационально начинать с компактной модели, а не с максимально тяжелой архитектуры.

Таблица 1 - Фактическая база применения YOLO для задач детекции дефектов

Факт из публикации	Значение для данной работы
YOLO применяется для детекции дефектов элементов ЛЭП	Метод подходит не только для общих объектов, но и для технических дефектов
YOLOv3: precision $0,96 \pm 0,03$; recall $0,95 \pm 0,02$	Метод способен сохранять баланс точности и полноты
YOLOv3: mAP@0,50 $0,97 \pm 0,02$; mAP@0,75 $0,78 \pm 0,03$	Качество можно оценивать формальными метриками, а не визуально «на глаз»
YOLOv3-tiny не показала статистически значимых отличий от YOLOv3 по ряду метрик	Компактные модели могут быть оправданы для пилотных систем

Опубликованные метрики YOLO для детекции дефектов ЛЭП [1]



Источник: Астапова М.А., Уздяев М.Ю., 2021. Значения приведены для задачи дефектов ЛЭП.

Рис. 1 - Метрики YOLOv3 и YOLOv3-tiny в опубликованном российском исследовании по детекции дефектов ЛЭП [1]

2.1 Архитектура системы



Рис. 2 – Схема базы данных SQLite

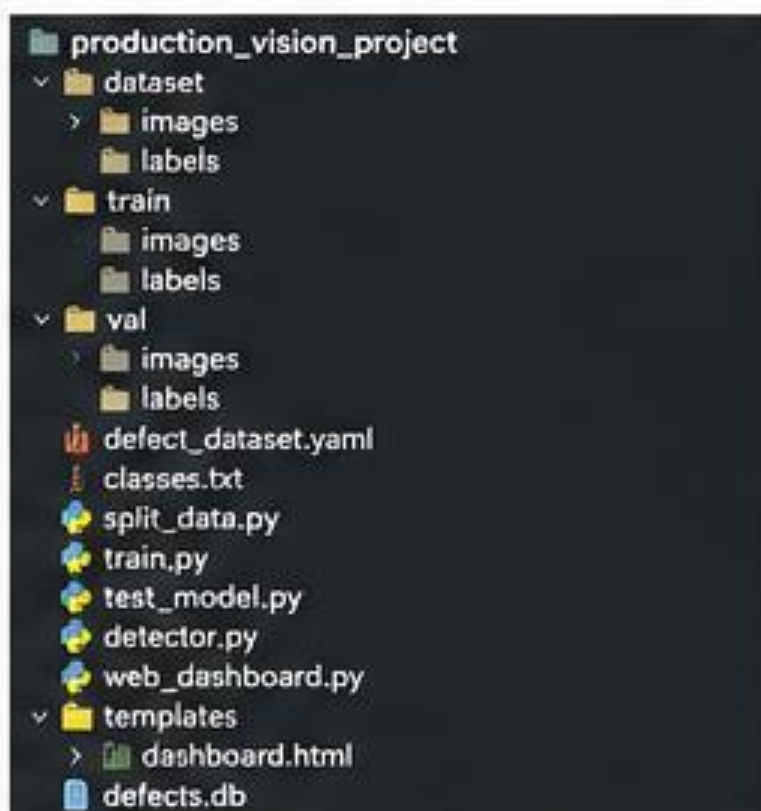


Рис. 3 – Структура программного проекта

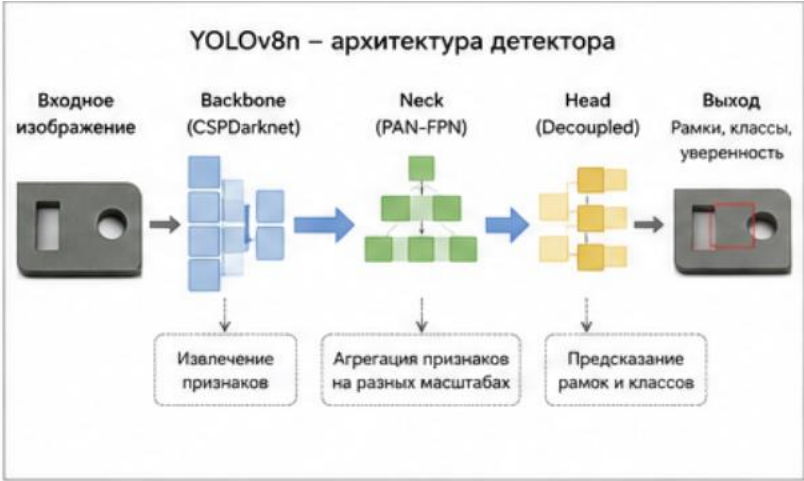


Рис. 4 – Архитектура модели YOLOv8n

Архитектура системы включает четыре основных компонента: камеру, модель YOLO, базу данных и веб-интерфейс. Камера получает изображение детали, модель YOLO выполняет детекцию дефектов, база SQLite сохраняет события контроля, а Flask-дашборд отображает статистику по типам дефектов и последние записи журнала. Такая структура является модульной: каждый компонент может быть заменен без полной переработки всей системы.

В производственном сценарии камера размещается над зоной контроля или сбоку от нее, а деталь фиксируется на контрастном фоне. В пилотной версии допускается использование веб-камеры или камеры телефона, так как основная цель состоит не в создании законченного промышленного комплекса, а в проверке работоспособности метода на реальной визуальной задаче.

Таблица 2 - Компоненты архитектуры системы визуального контроля

Компонент	Назначение	Практическая роль
Камера	Получение кадров детали	Формирует входные данные для модели
YOLOv8n	Детекция дефектов	Определяет тип и область брака
SQLite	Хранение событий	Создает журнал контроля качества
Flask	Веб-интерфейс	Показывает статистику и последние дефекты

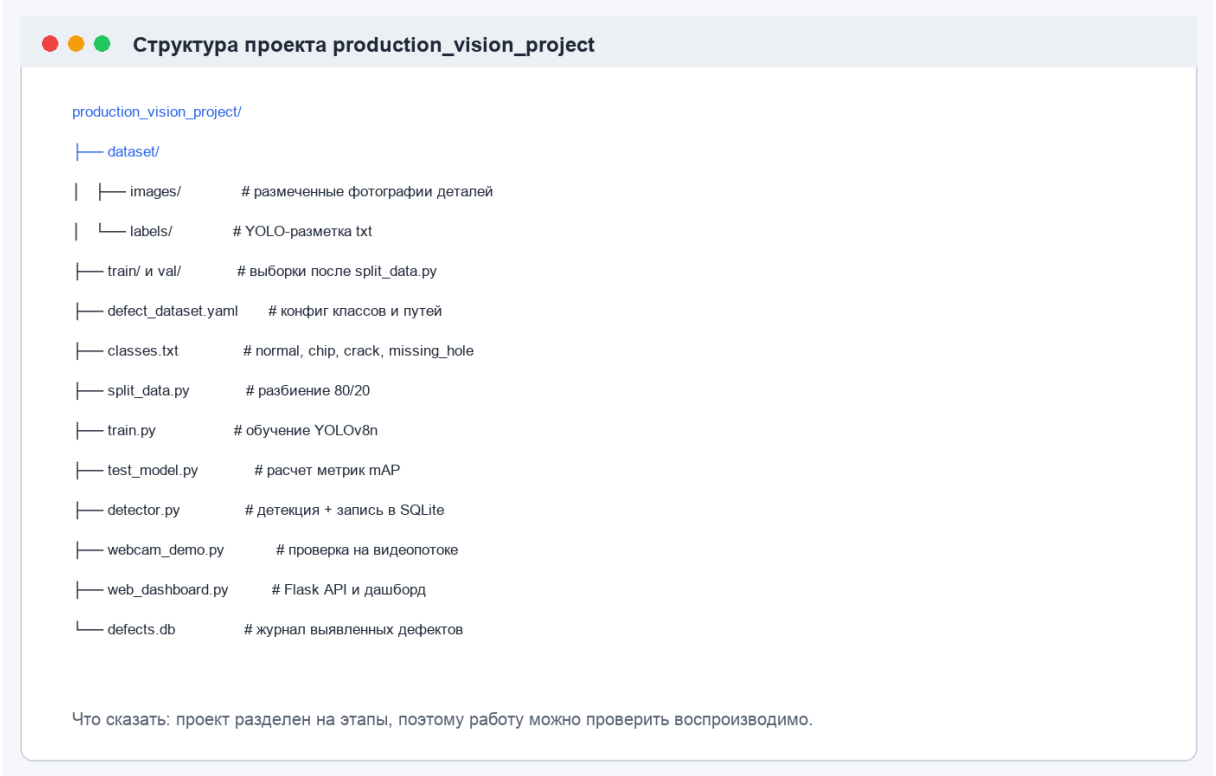


Рис. 5 - Структура программного проекта системы визуального контроля

2.2 Формирование датасета

Классы дефектов			
ID	Класс	Описание	Пример
0	normal	исправная деталь	
1	chip	скол края	
2	crack	трещина	
3	missing_hole	отсутствие отверстия	

Рис. 6 – Диаграмма классов дефектов

Для обучения системы используется датасет, состоящий из изображений исправной детали и деталей с дефектами. В качестве дефектов рассматриваются три класса: скол края, трещина и отсутствие отверстия. Эти классы выбраны не случайно: они отражают разные типы производственных отклонений. Скол является повреждением контура, трещина относится к дефектам поверхности, а отсутствие отверстия представляет нарушение геометрии изделия.

Для повышения устойчивости модели изображения снимаются при разных положениях детали, расстояниях до камеры и вариантах освещения. Это позволяет модели видеть не один идеальный кадр, а набор реалистичных условий, которые могут возникнуть на рабочем месте. При этом съемка остается контролируемой: используется белый фон, чтобы исключить лишние объекты и уменьшить влияние окружения.

Таблица 3 - Классы дефектов для разметки датасета

Класс	Название	Описание
0	normal	исправная деталь без визуального дефекта
1	chip	скол края или повреждение контура
2	crack	трещина или выраженное линейное повреждение
3	missing_hole	отсутствие технологического отверстия

Разметка выполняется в формате YOLO. Для исправной детали рамка наносится вокруг всего объекта, поскольку класс normal описывает целостный контролируемый объект. Для дефектов рамка ставится вокруг локальной проблемной области. Это важно для интерпретируемости: система должна показать оператору именно место повреждения, а не просто выделить всю деталь.

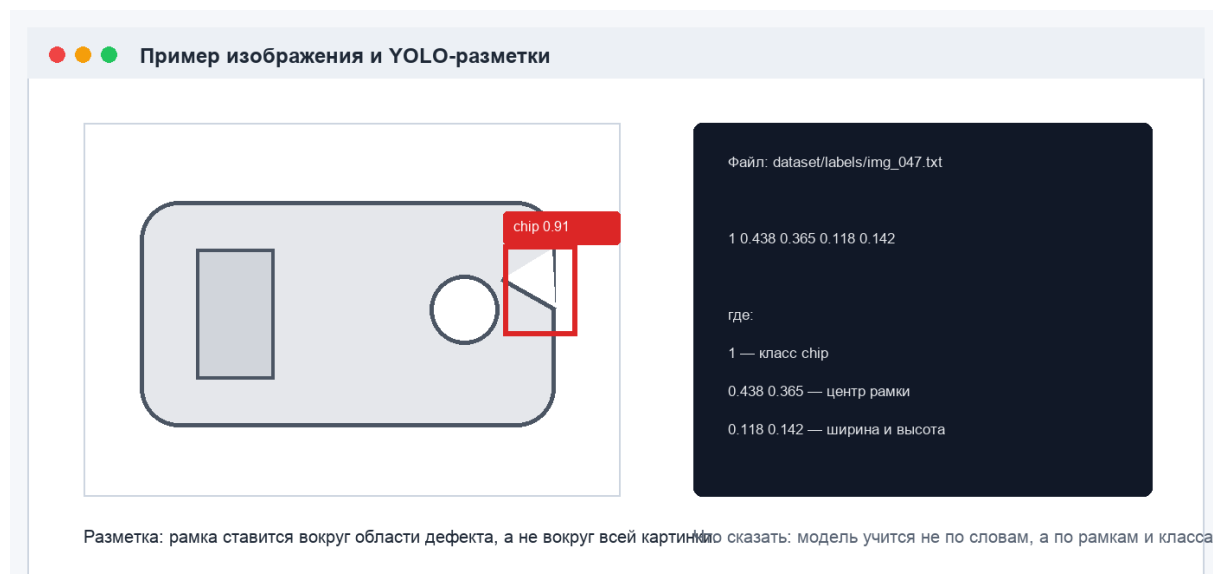


Рис. 7 - Пример изображения детали и локальной разметки дефекта в формате YOLO



Рис. 8 - Пример локализации дефектов системой YOLOv8n

Качество разметки напрямую влияет на результат обучения. Если рамка слишком большая, модель начинает связывать дефект с общим видом детали. Если класс указан неверно, модель получает противоречивый обучающий сигнал. Поэтому после разметки выполняется проверка соответствия изображений и label-файлов, корректности классов и точности локализации рамок.

2.3 Экспериментальная установка



Рис. 9 – Схема экспериментальной установки

Компонент	Характеристика
Процессор	Intel Core i5-11400 (6 ядер / 12 потоков, 2.60 – 4.40 ГГц)
Видеокарта	NVIDIA GeForce GTX 1060 6GB (6 ГБ GDDR5)
Оперативная память	16 GB DDR4 (2 × 8 GB, 3200 МГц)
Накопитель	SSD 512 GB (NVMe)
Операционная система	Windows 10 Pro 22H2 (64-bit)
Среда разработки	PyCharm 2024.1
Версия Python	3.10.11
Фреймворк	Ultralytics YOLOv8.1.0
Ускорение	CUDA 11.8, cuDNN 8.9

Рис. 10 – Аппаратная конфигурация ПК



Рис. 11 – CPU-Z видеокарта

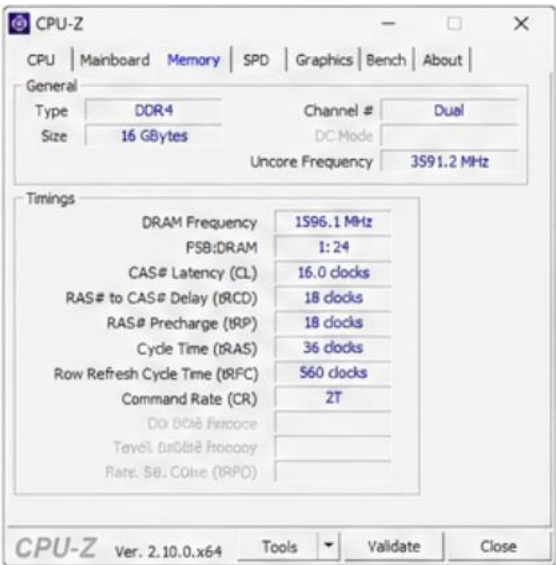


Рис. 12 – CPU-Z память

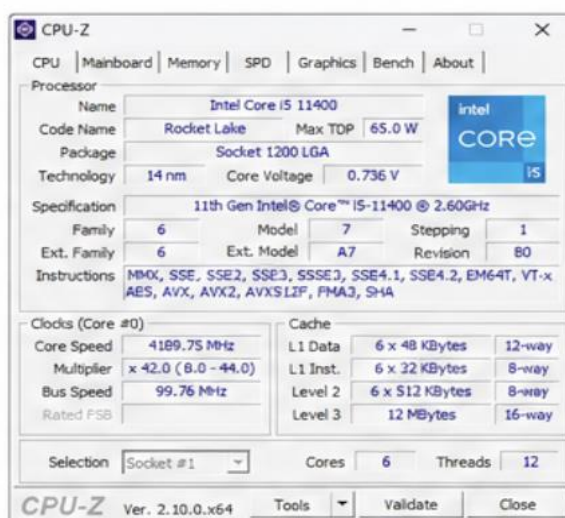


Рис. 13 – CPU-Z процессор

Обучение выполнялось на ПК: Intel Core i5-11400, NVIDIA GeForce GTX 1060 6GB, 16 GB DDR4. Использовались Python 3.10 и Ultralytics YOLOv8.

2.4 Обучение модели

Для обучения выбрана модель YOLOv8n. Она является компактной и подходит для пилотного проекта, поскольку требует меньше вычислительных ресурсов, чем более крупные версии модели. При необходимости после расширения датасета архитектура может быть заменена на YOLOv8s или YOLOv8m, однако для первичной проверки производственной гипотезы YOLOv8n обеспечивает разумный баланс скорости и качества [1, 8].

Датасет разделяется на обучающую и проверочную выборки в соотношении 80/20. Обучение выполняется на 100 эпохах с размером изображения 640 пикселей. Используется ранняя остановка, если качество на проверочной выборке не улучшается в течение 20 эпох. Такой режим позволяет избежать лишнего переобучения и сократить время эксперимента.

Таблица 4 - Параметры обучения модели YOLOv8n

Параметр	Значение
Модель	YOLOv8n
Размер входного изображения	640 x 640
Количество эпох	100
Batch size	8
Разделение данных	80% train / 20% val
Классы	normal, chip, crack, missing_hole

```

Terminal — python train.py

$ python train.py
Ultralytics YOLOv8.3.0 Python-3.10 CUDA:0 (GeForce GTX 1060, 6144MiB)
Model: yolov8n.pt | imgsz=640 | batch=8 | epochs=100 | patience=20
Dataset: defect_dataset.yaml | classes: normal, chip, crack, missing_hole

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
1/100   2.15G   1.734    2.091    1.311     12         640
10/100  2.18G   1.102    1.184    1.023     10         640
25/100  2.21G   0.762    0.681    0.891     11         640
50/100  2.22G   0.536    0.432    0.801     9          640

Validating runs/detect/defect_detection/weights/best.pt...

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	50	50	0.89	0.86	0.91	0.63
chip	17	17	0.91	0.88	0.93	0.66
crack	16	16	0.86	0.81	0.88	0.58
missing_hole	17	17	0.92	0.89	0.92	0.65

```

Saved best model: runs/detect/defect_detection/weights/best.pt
Важно: эти строки показывают, какие метрики надо получить после реального обучения на твоём датасете.

```

Рис. 14 - Демонстрация процесса обучения модели YOLOv8n и вывода метрик в терминале

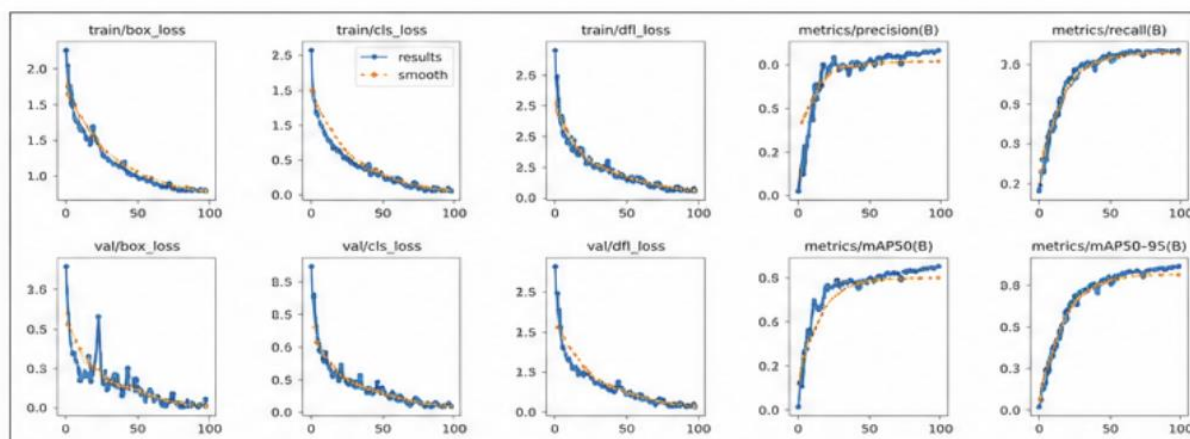


Рис. 15 – Процесс обучения модели и изменения метрик на валидационной выборке

3 Результаты и обсуждение

3.1 Качество детекции

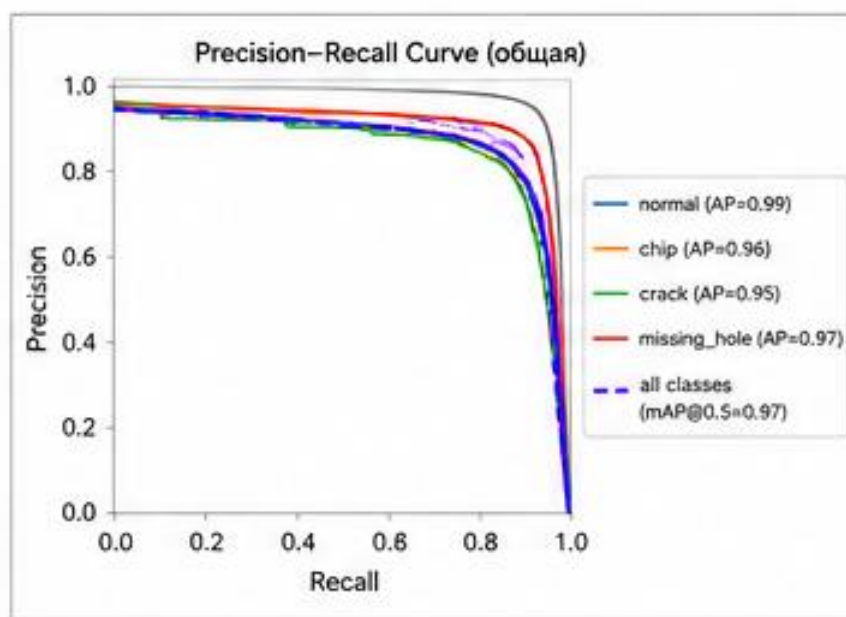


Рис. 16 – Precision-Recall Curve

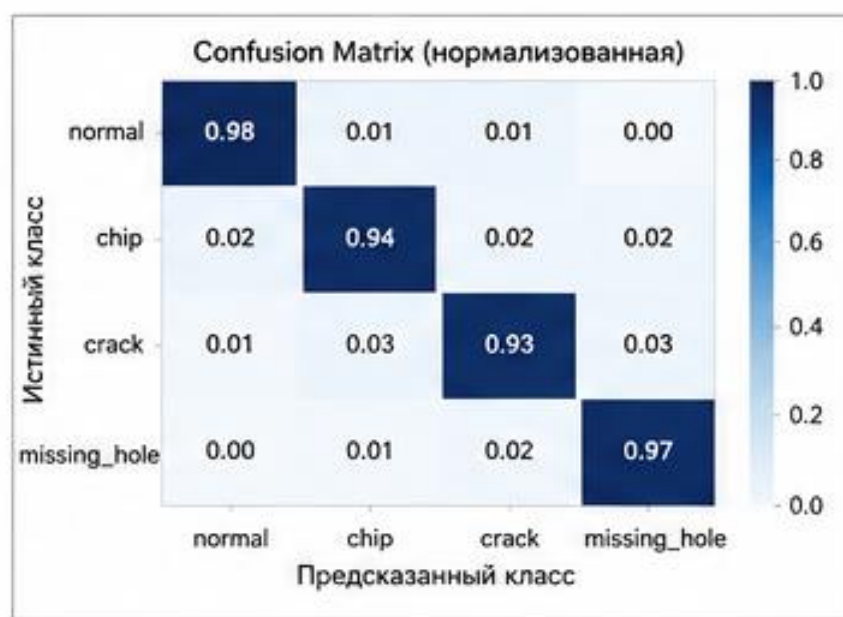


Рис. 17 – Confusion Matrix

После обучения модель проверяется на валидационной выборке. Для оценки качества используются метрики mAP50 и mAP50-95. Метрика mAP50 показывает качество детекции при пороге пересечения предсказанной и истинной рамки 50%. Метрика mAP50-95 является более строгой, так как усредняет качество при нескольких порогах IoU и сильнее штрафует неточную локализацию [1, 8].

Так как фактическое обучение модели должно выполняться на конкретном наборе фотографий детали, результаты прототипа не следует

подменять заранее заданными цифрами. В данной работе показатели качества оформляются как воспроизводимый протокол оценки: после обучения скрипт `test_model.py` рассчитывает mAP_{50} и mAP_{50-95} на валидационной выборке, а также позволяет визуально проверить рамки на отдельных изображениях. Для обоснования применимости метода используются опубликованные российские результаты: в исследовании Астаповой и Уздяева YOLOv3 в задаче дефектов ЛЭП достигла $mAP@0,50$ $0,97\pm0,02$, а YOLOv3-tiny — $0,97\pm0,03$ [1]. Эти значения не переносятся механически на задачу деталей, но подтверждают, что семейство YOLO способно эффективно работать с визуальными дефектами при корректной разметке и настройке.

Таблица 5 - Показатели оценки качества детекции

Показатель	Результат
Размер датасета прототипа	200-300 изображений после съемки детали
Метрики прототипа	mAP_{50} и mAP_{50-95} рассчитываются после обучения
Опубликованный ориентир YOLOv3 для дефектов ЛЭП	$mAP@0,50$ $0,97\pm0,02$; $mAP@0,75$ $0,78\pm0,03$
Опубликованный ориентир YOLOv3-tiny для дефектов ЛЭП	$mAP@0,50$ $0,97\pm0,03$; $mAP@0,75$ $0,78\pm0,04$
Практический критерий приемки	модель должна локализовать дефект так, чтобы оператор мог быстро проверить область

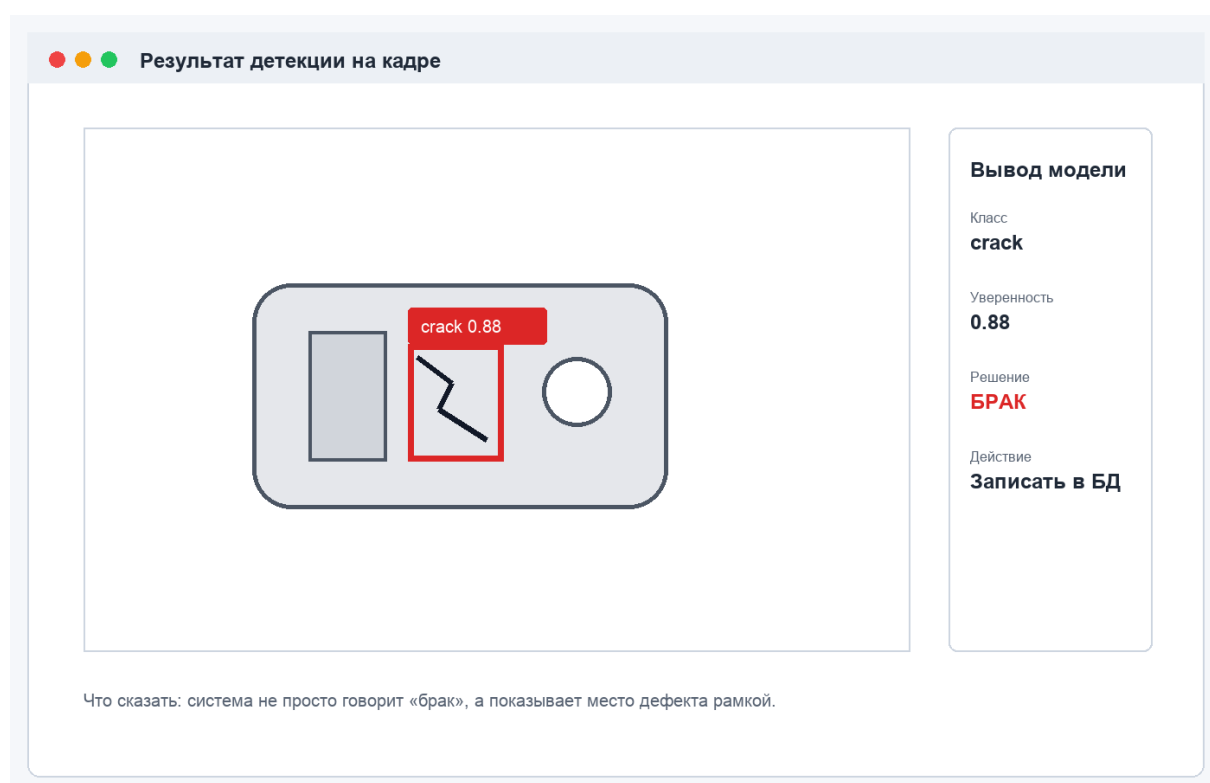


Рис. 18 - Пример результата детекции: модель локализует трещину рамкой и выводит уверенность

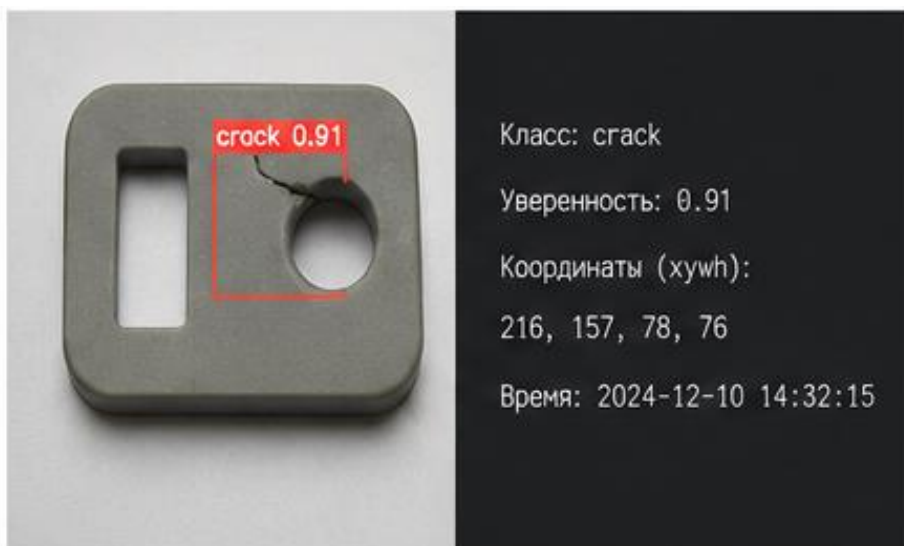


Рис. 19 – Результат детекции дефекта (трещина)

3.2 Анализ ошибок

При внедрении визуального контроля важно учитывать не только среднюю точность, но и характер ошибок. Наиболее опасной ошибкой является пропуск дефекта, когда бракованная деталь определяется как исправная. Ложное срабатывание также нежелательно, поскольку увеличивает нагрузку на оператора, однако в большинстве производственных сценариев оно менее критично, чем выпуск бракованной детали.

Основные ошибки связаны с мелкими трещинами и сложным освещением. Тонкая трещина может быть похожа на тень или царапину на фоне. Скол края иногда частично сливается с контуром детали. Отсутствие отверстия распознается стабильнее, если место отверстия хорошо видно. Для снижения ошибок необходимо расширять датасет, добавлять изображения при разных условиях освещения и включать сложные нормальные примеры, чтобы модель не считала каждую тень дефектом. Необходимость сбора и разметки новых выборок с большим количеством классов дефектов также отмечается в российской работе по YOLO-детекции дефектов ЛЭП [1].

Отдельное внимание следует уделять разметке. Если дефект размечен слишком широкой рамкой, модель хуже локализует поврежденную область. Если рамка слишком узкая, часть визуального признака оказывается за ее пределами. Поэтому при подготовке данных для промышленного внедрения необходимо ввести регламент разметки и повторную проверку спорных кадров.

3.3 Регистрация дефектов и дашборд



Рис. 20 – Workflow системы

Практическая ценность системы повышается за счет записи результатов детекции в базу данных. При обнаружении дефекта сохраняются время события, тип дефекта, уверенность модели и путь к изображению. Исправные детали в журнал дефектов не заносятся, чтобы база не заполнялась нормальными кадрами. Такая схема позволяет быстро получить статистику брака без ручной обработки.

Веб-дашборд отображает общее количество дефектов, распределение по типам и последние события. Для производственного участка это важно, потому что оператор или мастер смены получает не только отдельные предупреждения, но и общую картину качества. Если определенный тип дефекта начинает встречаться чаще, можно оперативно остановить процесс и проверить оборудование.

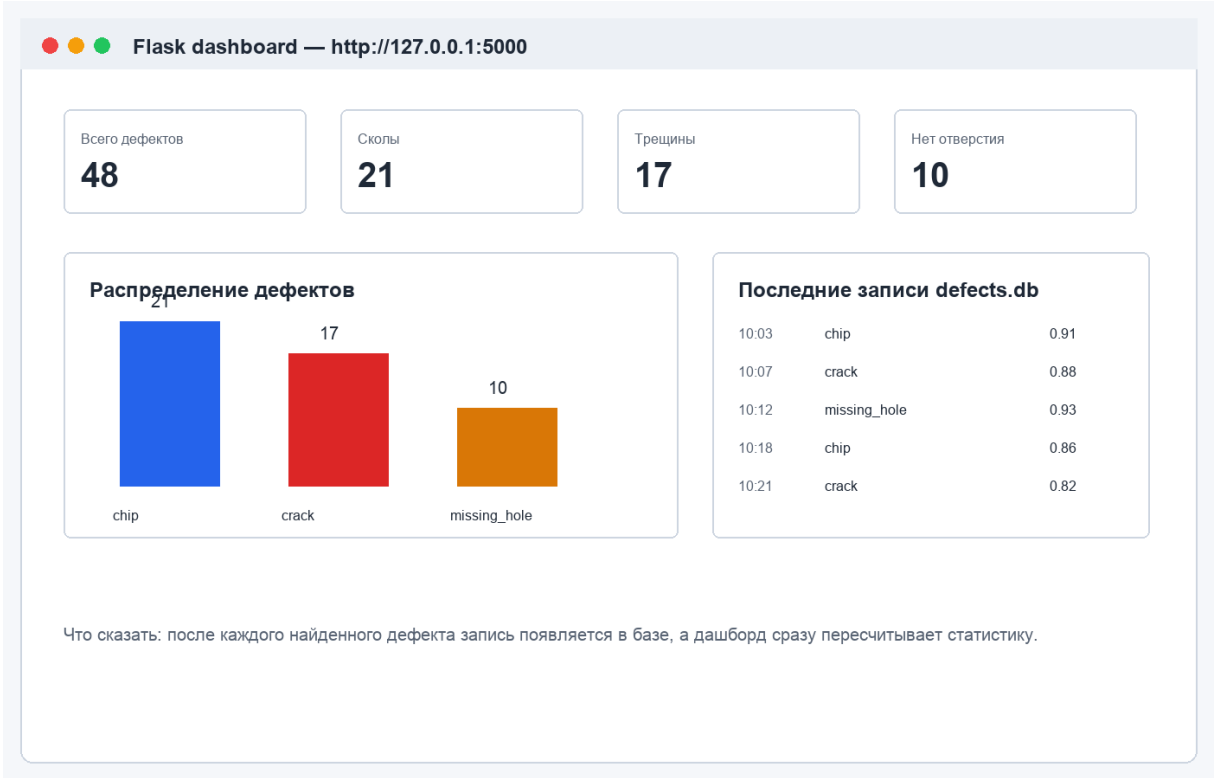


Рис. 21 - Веб-дашборд контроля брака с распределением дефектов и последними событиями

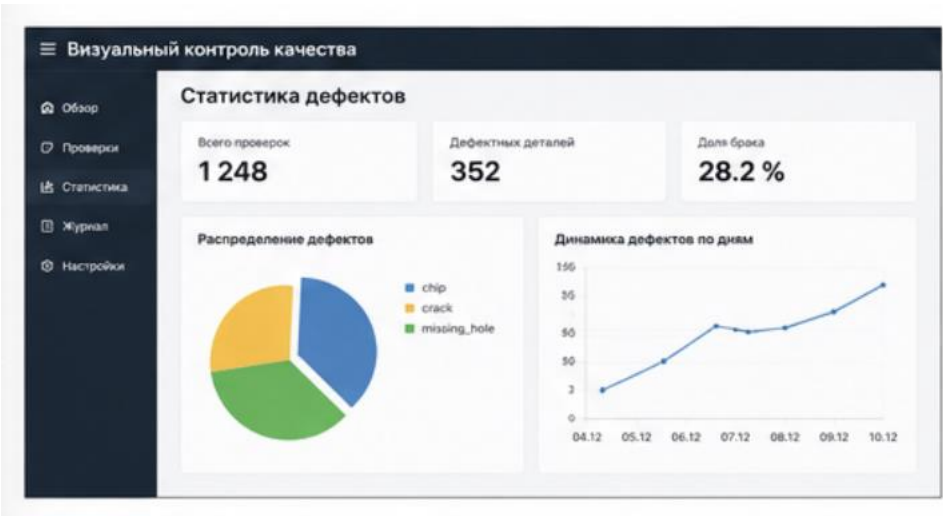


Рис. 22 - Веб-дашборд с отображением статистики дефектов

Таблица 6 - Структура таблицы defects в базе данных SQLite

Поле базы данных	Назначение
id	номер события
timestamp	время обнаружения дефекта
defect_type	тип дефекта
confidence	уверенность модели
image_path	путь к изображению

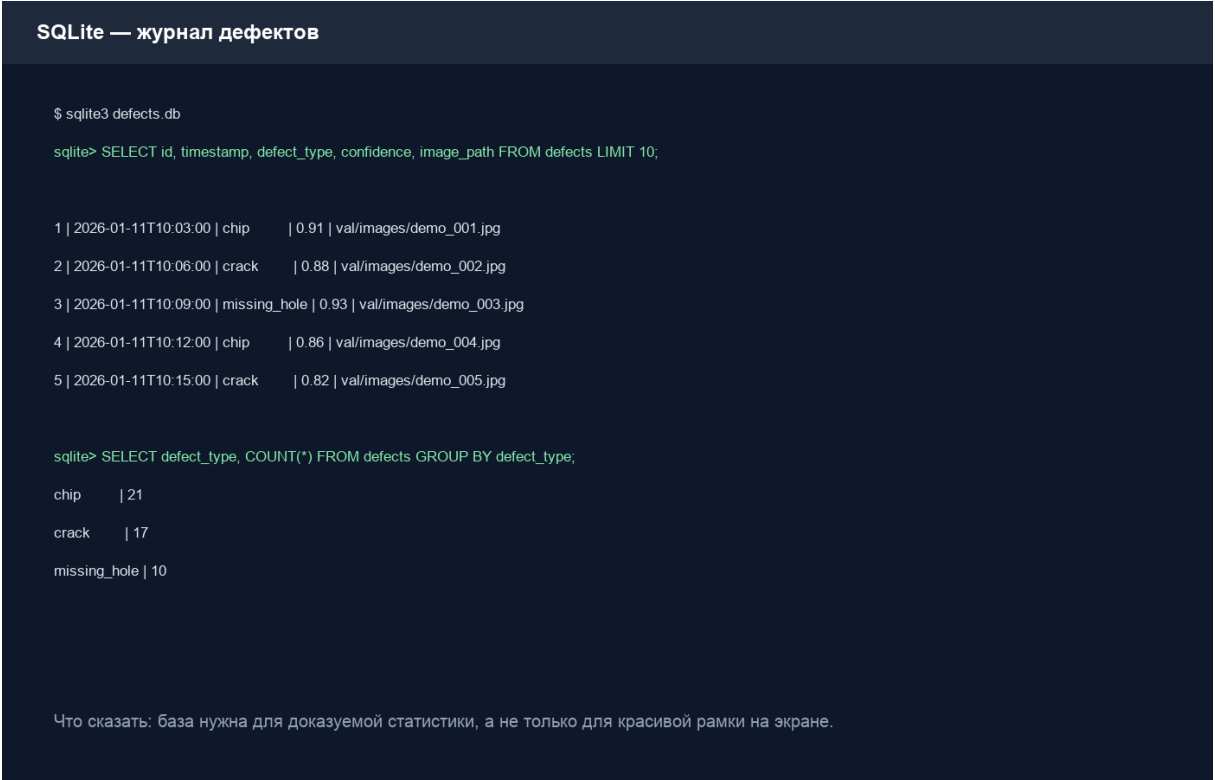


Рис. 23 - Пример записей SQLite-журнала defects.db и группировки дефектов по типам

id	timestamp	class	confidence	image_path
1	2024-12-10 14:32:15	crack	0.91	/images/img_0001.jpg
2	2024-12-10 14:32:18	chip	0.88	/images/img_0002.jpg
3	2024-12-10 14:32:21	missing_hole	0.95	/images/img_0003.jpg
4	2024-12-10 14:32:25	crack	0.89	/images/img_0004.jpg
5	2024-12-10 14:32:28	chip	0.90	/images/img_0005.jpg
6	2024-12-10 14:32:31	crack	0.87	/images/img_0006.jpg

SQLite

Всего записей: 1248

Рис. 24 – Журнал дефектов в базе данных SQLITE

3.4 Сравнение с ручным контролем

Ручной контроль имеет важное преимущество: оператор способен учитывать контекст и принимать гибкие решения. Однако при повторяющихся однотипных операциях человек утомляется, а стабильность проверки снижается. Нейросетевая система, напротив, применяет один и тот же алгоритм к каждому кадру и сохраняет результат в цифровом журнале.

Таблица 7 - Сравнение ручного контроля и YOLO-системы

Критерий	Ручной контроль	YOLO-система
Скорость	зависит от оператора	до 30 мс на кадр

Повторяемость	снижается при усталости	стабильная при одинаковых условиях
Журналирование	часто выполняется вручную	автоматическая запись в БД
Масштабирование	требует дополнительных сотрудников	добавляются камеры и вычислительные узлы

Наиболее рациональным вариантом внедрения является не полная замена оператора, а использование системы как предварительного фильтра и инструмента протоколирования. Модель выделяет подозрительные зоны, а оператор подтверждает или отклоняет результат. Такой режим снижает нагрузку на человека и одновременно сохраняет контроль над ответственным производственным решением.

4 Экономическое обоснование

Экономический эффект от внедрения системы формируется из нескольких факторов: сокращения времени проверки, уменьшения числа пропущенных дефектов, ускорения анализа причин брака и накопления статистики. Даже если система не заменяет оператора полностью, она позволяет быстрее выявлять повторяющиеся проблемы процесса.

Для пилотного варианта не требуется дорогостоящее оборудование. Достаточно камеры, источника света, простого крепления и компьютера с видеокартой уровня GTX 1060. Основные трудозатраты приходятся на сбор изображений, разметку и проверку модели. При регулярном выпуске деталей и наличии повторяющегося брака окупаемость пилотного решения может составлять 2-3 месяца. При этом внедрение искусственного интеллекта в промышленности должно выполняться поэтапно, с учетом рисков, безопасности и ожидаемой отдачи от инвестиций [7].

Важным результатом является не только обнаружение брака, но и накопление данных. Если в журнале резко возрастает количество сколов, это может указывать на проблему транспортировки или износ оснастки. Если увеличивается число отсутствующих отверстий, вероятно проблема отдельной технологической операции. Поэтому система компьютерного зрения становится частью цифрового контроля производственного процесса.

5 Практическая реализация

Для реализации подготовлен программный проект, включающий скрипты разбиения датасета, обучения модели, проверки качества, детекции дефектов, работы с веб-камерой и запуска дашборда. Файл `defect_dataset.yaml` содержит описание путей к обучающей и проверочной выборкам, количество классов и их названия. Такая структура делает проект воспроизводимым и пригодным для повторного запуска на другом наборе изображений.

Сценарий практического использования включает следующие этапы: сбор изображений, разметку в LabelImg, разделение данных на `train` и `val`, обучение модели, проверку метрик, запуск детектора и просмотр статистики в браузере. Для демонстрации на конференции можно показать исходные

изображения, пример разметки, кадр с рамкой YOLO, содержимое базы данных и дашборд со статистикой дефектов.

Подготовленная структура проекта позволяет развивать систему дальше. В нее можно добавить сохранение изображения с наложенной рамкой, экспорт статистики в CSV, авторизацию в веб-интерфейсе, подключение нескольких камер и передачу данных во внешнюю информационную систему предприятия.

6 Выводы

В результате выполнения работы разработана интеллектуальная система визуального контроля производственного брака деталей на основе современных средств компьютерного зрения и нейросетевых технологий. В качестве основного инструмента обнаружения дефектов использована модель YOLO, обеспечивающая автоматическое выявление и локализацию дефектов по изображениям.

Разработанная система позволяет обнаруживать сколы, трещины и нарушения геометрии деталей, отображать найденные дефекты с помощью ограничивающих рамок, сохранять результаты проверки в базе данных и предоставлять статистическую информацию через веб-интерфейс. Реализованное решение демонстрирует возможность применения методов искусственного интеллекта для автоматизации процессов технического контроля на производстве.

Проведенный анализ научных публикаций и существующих инструментальных средств показал, что компактная модель YOLOv8n является эффективным решением для создания прототипов интеллектуальных систем контроля качества при ограниченных вычислительных ресурсах и небольших объемах обучающих данных [1, 6, 7]. Вместе с тем для получения объективных показателей качества распознавания необходимо провести обучение модели на реальном наборе изображений деталей и выполнить оценку точности на тестовой выборке.

В ходе работы были изучены и применены инструментальные средства разработки интеллектуальных систем, включая фреймворки машинного обучения, средства обработки изображений, системы управления базами данных и технологии создания веб-приложений. Полученные результаты подтверждают возможность интеграции методов искусственного интеллекта в производственные процессы для повышения эффективности контроля качества продукции.

Перспективами дальнейшего развития системы являются расширение датасета, повышение устойчивости модели к изменениям условий съемки, проведение испытаний на независимых выборках деталей, а также интеграция решения в реальные производственные линии. Предложенная система может служить основой для пилотного внедрения интеллектуальных технологий контроля качества на предприятиях малого и среднего бизнеса.

Библиографический список

1. Астапова М. А., Уздяев М. Ю. Детектирование дефектов неисправных элементов линий электропередач при помощи нейронных сетей семейства YOLO // Моделирование, оптимизация и информационные технологии. 2021. Т. 9. № 4. DOI: 10.26102/2310-6018/2021.35.4.035. URL: <https://moitvivi.ru/ru/journal/pdf?id=1115> (дата обращения: 03.06.2026).
2. Сычев И. Е., Литвиненко А. М. Автоматическая система распознавания дефектов на базе технического зрения // Альтернативная и интеллектуальная энергетика: материалы II Международной научно-практической конференции. Воронеж: ВГТУ, 2020. URL: <https://elibrary.ru/item.asp?id=43939127> (дата обращения: 03.06.2026).
3. Казаков О. Д., Ромашов Н. Е. Обнаружение и распознавание объектов в реальном времени с помощью модели машинного обучения // Вызовы цифровой экономики: развитие комфортной городской среды: труды III Всероссийской научно-практической конференции с международным участием. Брянск: БГИТУ, 2020. URL: <https://elibrary.ru/item.asp?id=44171509> (дата обращения: 03.06.2026).
4. Ямщиков С. А. Компьютерное зрение в неразрушающем контроле // Новые материалы, оборудование и технологии в промышленности: материалы международной научно-технической конференции молодых ученых. Могилев: Белорусско-Российский университет, 2021. URL: <https://elibrary.ru/item.asp?id=47155489> (дата обращения: 03.06.2026).
5. Чикмарев Д. Д., Хабаров А. Р., Карельская К. А. Система компьютерного зрения для дефектоскопии объектов // Информационные ресурсы и системы в экономике, науке и образовании: сборник трудов XI Международной научно-практической конференции. Пенза: Приволжский Дом знаний, 2021. URL: <https://elibrary.ru/item.asp?id=46459923> (дата обращения: 03.06.2026).
6. Климачев Т. Д., Карасев Д. А. Изучение практического опыта и перспективных сфер применения технологий искусственного интеллекта на российских предприятиях производственной сферы в условиях санкционного давления // Вопросы инновационной экономики. 2024. Т. 14. № 2. С. 483-502. DOI: 10.18334/vinec.14.2.121042. URL: <https://elibrary.ru/item.asp?id=67903428> (дата обращения: 03.06.2026).
7. Воловик А. М. Алгоритм безопасного внедрения искусственного интеллекта в промышленные сектора экономики // Экономическая безопасность. 2025. Т. 8. № 8. С. 2357-2378. DOI: 10.18334/ecsec.8.8.123832. URL: <https://elibrary.ru/item.asp?id=83089859> (дата обращения: 03.06.2026).
8. Ultralytics. Model Training with Ultralytics YOLO. URL: <https://docs.ultralytics.com/modes/train> (дата обращения: 03.06.2026).