

Разработка программных средств подготовки данных для систем компьютерного зрения в серийном производстве¹

В. А. Зотов*, Е. Ю. Печенина** В. А. Печенин***

* *Передовая инженерная аэрокосмическая школа, Самарский университет, Самара*

** *Кафедра технологий производства двигателей, Самарский университет, Самара*

*** *Научно-исследовательская лаборатория 'Искусственный интеллект в производственных системах', Самарский университет, Самара*

Поступила в редколлегию 16.10.2025 г. Принята 25.11.2025 г.

Аннотация—Аннотация— Разработано приложение для разметки видеоматериала, позволяющего в условиях производства, ограниченного количества и широкой номенклатуры, автоматизировать разметку видеоматериала для дальнейшего обучения нейронных сетей, используемых в системах компьютерного зрения. Для достижения достаточной точности при сравнительно небольшом объёме обучающей выборки, разработан каскадных алгоритм идентификации, использующий несколько нейронных сетей для разметки. При тестировании приложения значение метрики $mAP_{\{50-95\}}$ составила 0.93.

КЛЮЧЕВЫЕ СЛОВА: компьютерное зрение, нейронная сеть, алгоритм, детектирование, классификация.

DOI: 10.53921/18195822_2025_25_3.1_608

1. ВВЕДЕНИЕ

На современных производствах остро стоят вопросы автоматизации контроля качества продукции и самих процессов. Для решения первой группы вопросов внедряется оборудование с числовым программным управлением и разрабатываются специальные алгоритмы для них [1, 2]. Для решения вопросов второй группы используются системы сбора информации и мониторинга (MDC-системы) и системы оперативного учёта выполнения производственных заданий (MES-системы) [3].

В настоящее время, наиболее надёжным инструментом для детектирования объектов при наличии «информационного шума» являются глубокие нейронные сети [4, 5]. На промышленных предприятиях с широкой номенклатурой производства (300-400 типов деталей на одном участке) требуется достаточно быстрый (быстрее человека) и надёжный (допускающий меньше ошибок) алгоритм, позволяющий идентифицировать производственный передел (детали, сборочные единицы, тары). Результаты точности работы одной нейронной сети, например YOLO, на такой малой выборке могут не удовлетворять подобным требованиям, в тоже время сильно обгоняя по скорости человека [6]. Однако YOLO, обученная детектировать конкретный класс (например, QR-метки) позволяет добиваться высокой точности работы в условиях производства и относительно малых обучающих выборок, хотя скорость обработки данных при таком подходе возрастает в разы. Использование нескольких нейронных сетей, работающих

¹ Работа выполнена по проекту FSSS-2024-0019, реализуемого в рамках федерального проекта «Развитие человеческого капитала в интересах регионов, отраслей и сектора исследований и разработок», результат «Созданы новые лаборатории, в том числе под руководством молодых перспективных исследователей».

последовательно над задачей детектирования объектов производства, можно использовать для создания инструмента автоматизации разметки видеоматериала для последующего обучения нейронной сети, работающей быстрее в условиях работы стриминговых систем. Целью работы являлось создание программного приложения, позволяющего выполнять автоматизированную разметку данных для обучения нейронных сетей для систем компьютерного зрения. Для автоматизации разметки использован комплекс обученных на сравнительно небольшой выборке данных нейронных сетей, объединённых в каскадный алгоритм.

2. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

Для прослеживаемости объектов в производстве написаны стандарты [7], целью внедрения которых является уменьшение ошибок при сборе необходимой информации и сокращение времени сбора за счёт использования средств автоматизации, в частности идентифицирующих меток (QR-кодов, ArUco-маркеров). Детали движутся в тарах, и организация адекватного отслеживания материальных потоков по цепочке технологического передела должна позволять детектировать местоположение каждой детали в составе партии. Для реализации такого подхода необходимо соблюдать принцип однозначного соответствия «ID детали -ID тары -ID ячейки в таре». В связи с описанным выше, с использованием алгоритмов машинного обучения нужно детектировать метки на таре, её ячейки и классифицировать содержимое каждой ячейки.

2.1. Используемые модели нейронных сетей и общий алгоритм обучения

Для работы бизнес-логики приложения разработан каскадный алгоритм, основная идея которого заключается в использовании одной нейронной сети YOLO [6, 8] для детектирования конкретного класса – ArUco, QR, ячейки конкретной тары. Для повышения точности классификации объектов внутри ячеек была использована дополнительная нейронная сеть VGG19. Этот шаг позволит уменьшить количество ошибок второго рода (ложное заключение об отсутствии объекта) и повысить mAP [9] результата. На рис. 1 приведена блок-схема этапов разработанного алгоритма.

На третьем шаге с использованием нейронной сети YOLO выполняется детектирование N ячеек тары. Все детектированные фрагменты, содержащие ячейки, проходят в цикле через нейронную сеть для классификации VGG. Перед классификацией происходит изменение размера, и, при необходимости, цветового режима. В результате работы всего алгоритма формируется информация об объекте и ограничивающем его прямоугольнике в виде списка словарей формата: 'number of frame': ['type of object', 'degree of confidence', 'center of x', 'center of y', 'width', 'height']. Таким образом, формируются необходимые данные для разметки.

2.2. Архитектура приложения

Приложение разрабатывалось в объектно-ориентированной парадигме. На рис. 2 приведена схема взаимодействия классов приложения.

Класс MainWindow отвечает за функционирование главного окна программы, содержит виджеты и принимает запросы от пользователя. Из него можно обратиться в окно настроек, функционал которого реализован в классе Settings. Информационные сообщения об ошибках и окончании процесса появляются в дополнительном окне, за которое отвечает класс MessageWindow. Соответственно в MainWindow создаются экземпляры классов дополнительных окон.

Работа с видео и изображениями производится в классах, объединённых на схеме рисунка 1, названием players. Это два класса - VideoPlayer и ImagePlayer. Их функционал – работа с



Рис. 1. Каскадный алгоритм для детектирования объектов

видео и изображениями, выполнение разметки и сохранение результатов в JSON файл. Функционал по разметке с использованием нейронных сетей реализован в классах, объединенных в группу «detector» – ArUcoDetector, QRDetector и NNDetector. Каждый из классов отдельно отвечает за работу с ArUco маркерами, QR метками и детектированием ячеек с классификацией объектов внутри них. Пути к весам нейронных сетей хранятся в классах, объединенных в группу «...Constans», находящихся в директории core. Дополнительная обработка QR и ArUco реализована в классах группы «...handler».

2.3. Оценка точности

Для оценки точности результатов детектирования использовалась метрика $mAP_{\{50-95\}}$ (Mean Average Precision) являющаяся средним значением десяти метрик mAP, в свою очередь являющихся средним значением всех детектируемых классов на изображении для метрик AP (Average Precision) для каждого класса. Рассчитывается AP для различных порогов IoU (Intersection over Union) от 0,5 до 0,95 с шагом 0,05. В контексте детектирования объектов, IoU (Intersection over Union) — это метрика, используемая для оценки точности предсказания модели. Она измеряет пересечение между двумя прямоугольниками и делит его на объединение этих прямоугольников. Если IoU больше порога, то предсказанный объект истинно положительный

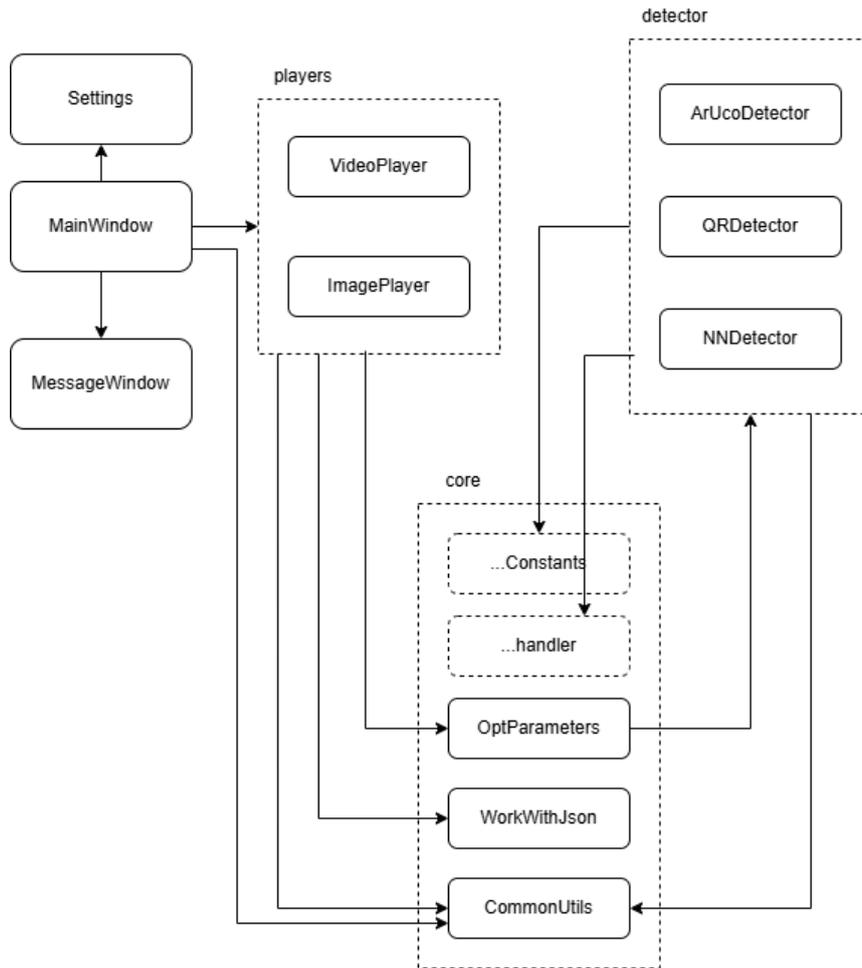


Рис. 2. Схема взаимодействия классов приложения

– TP. Если IoU меньше порога, то предсказанный объект ложно положительный – FP. Если присутствующий на изображении объект сеть не обнаружила, то он классифицируется как ложно отрицательный FN. Любая другая область изображения (не содержащая объект и где модель его не предсказала) может считаться истинно отрицательной TN. Точность работы ас для классификации (сети VGG19) оценивается как отношение количества правильно классифицированных изображений к их общему количеству. Функцией потерь для используемой в работе нейронной сети для классификации VGG19 является категориальная кросс-энтропия.

3. ЭКСПЕРИМЕНТЫ И РЕЗУЛЬТАТЫ

Для проведения экспериментов детали помещались в тару с обрешёткой, в каждой ячейке могла находиться одна деталь (или ничего). По углам тары нанесены маркеры – ArUco для автоматической активации систем компьютерного зрения и QR-код для анализа информации по таре в подобных системах. Необходимо выделить и классифицировать ячейки ящика как пустые или содержащие детали (дополнительно определив тип детали), а также детектировать QR- и ArUco-коды, расположенные на углах ящика.

3.1. Обучение детекторов YOLO

Была выполнена запись 6 видео: 5 обучающих и 1 проверочная, для обучения четырех сетей YOLO. Суммарно обучающая выборка содержит 1805 кадров, проверочная выборка составила 740 кадров. Разрешение каждого кадра было фиксировано: 640x480 пикселей.

Для детектирования ArUco маркеров, QR маркеров и ячеек были обучены отдельные нейронные сети YOLOv3, отвечающие за детектирование одного класса соответственно. Обучение проводилось с использованием видеокарты Tesla T4 с 16Гб видеопамяти, количество эпох обучения 2000. Для обучения сети детектирования ArUco использовалось 108 кадров из их общего количества. Для обучения сети детектирования QR кодов использовалось 137 кадров из их общего количества. На обучающих фотографиях присутствуют два типа тар – пластиковая и деревянная. Для обучения сети ячеек пластиковой тары использовалось 246 кадров из их общего количества. Для обучения сети детектирования ячеек деревянной тары использовалось 41 кадр из их общего количества.

В результате потребовалось до 300 изображений для каждой из нейронных сетей YOLO, использованных в приложении. Значение метрики $mAP_{\{50-95\}}$ составило от 0,868 до 1 для каждого типа детектирования (маркеры, ячейки).

3.2. Обучение классификаторов VGG

Стояла задача обучить шесть моделей VGG для классификации двух классов – пустых (класс #1) и занятых одним из шести типов деталей ячейках для двух типов тар (класс #2). Обучение проводилось с использованием видеокарты NVIDIA GeForce GTX 1080 Ti с 11Гб видеопамяти. В два этапа были подобраны оптимальные параметры обучения, которые в дальнейшем должны применяться при обучении новых моделей на новые типы деталей. В таблице 1 приведены данные по количеству изображений для обучающих и тестовых выборок по рассматриваемым деталям.

Таблица 1. КОЛИЧЕСТВО ИЗОБРАЖЕНИЙ ДЛЯ ОБУЧЕНИЯ И ТЕСТА

Деталь name	Обучающая выборка		Тестовая выборка	
	Класс 1	Класс 2	Класс 1	Класс 2
Part 1	638	638	67	67
Part 2	150	301	21	77
Part 3	174	451	29	67
Part 4	160	172	11	12
Part 5	606	606	67	67
Part 6	648	648	72	72

На первом этапе подбирались параметры: тип оптимизатора, количество эпох и размер батча. Для проведения экспериментов использовалась деталь номер 1. В качестве оптимизаторов были выбраны стохастический градиентный спуск (SGD)[10] и адаптивный метод оптимизации Adam [11].

В рамках исследования так же проводилось изучение влияния типа изображения (цветное или серое) на эффективность процесса обучения. Выдвигалась гипотеза о том, что использование серых изображений может способствовать сокращению времени обучения за счёт уменьшения количества обрабатываемых матриц чисел по сравнению с цветными изображениями. Но результаты экспериментов показали, что модели, обученные на цветных изображениях, демонстрируют более высокую точность по сравнению с моделями, обученными на серых изображениях. В дальнейшем все эксперименты проводились исключительно с использованием цветных изображений.

На основании анализа экспериментальных данных первого этапа был сужен диапазон поиска параметров для второго этапа: размер батча от 4 до 8, количество эпох 40–60 и оптимизатор стохастического градиентного спуска (SGD). Таким образом, общее количество возможных значений параметров второго этапа составило 9. Итоговые результаты обучения и тестирования нейронных сетей представлены в таблицах 2 и 3.

Таблица 2. Обобщенные результаты с параметрами наилучшего соответствия для обучающей выборки

Деталь	Размер батча	Количество эпох	Функция потерь	Точность
Part 1	8	40	0.00282	0.99913
	8	60	$5.73 \cdot 10^{-5}$	1
Part 2	6	60	0.00057	1
	8	40	0.00622	1
Part 3	4	40	0.07151	0.96975
	8	40	0.1001	0.96263
Part 4	6	60	$8.19 \cdot 10^{-5}$	1
	8	40	0.30114	0.97315
Part 5	4	60	$1.87 \cdot 10^{-5}$	1
	6	50	$6.07 \cdot 10^{-5}$	1
Part 6	8	40	0.00089	1
	8	60	$5.45 \cdot 10^{-5}$	1

Таблица 3. Общие результаты с наилучшим параметрическим выводом на тестовой выборке

Деталь	Размер батча	Количество эпох	Функция потерь	Точность
Part 1	8	40	0.67911	0.92537
	8	60	0.79422	0.94029
Part 2	6	60	0.4419	0.90816
	8	40	0.47636	0.88776
Part 3	4	40	1.01542	0.70833
	8	40	1.10272	0.69792
Part 4	6	60	0.00019	1
	8	40	0.49471	0.82609
Part 5	4	60	0.10271	0.98507
	6	50	0.17105	0.98507
Part 6	8	40	0.43632	0.93055
	8	60	0.86021	0.88889

Лучшие результаты дал размер батча 8 и количество эпох 40. Сбалансированность количества изображений двух классов так же влияет на точность, а кроме того размер обучающей выборки желательнее не ниже 1200 изображений. График точности и функции потерь детали номер 5, для которой эти параметры на тестовой выборке достигли лучших значений, представлен на рис. 3.

Результирующая точность детектирования ячеек дополнялась уверенностью в правильности классификации находящихся в них объектов с использованием нейронной сети VGG19. Метрика точности работы разработанного алгоритма (Рис. 1) $mAP_{\{50-95\}}$ на тестовой выборке из 740 кадров для всех классов составила 0.93. Для сравнения на обучающей выборке была обучена одна нейронная сеть YOLO, детектирующая все классы, ее точность составила 0.71. Интерфейс разработанного приложения с разметкой приведен на рис. 4.

В главном окне, сверху, располагаются: поле для ввода шага количества кадров, через который будет производится их сохранение из видео в отдельные изображения; опция «Сохранять фото» - выполнить нарезку фото из видео; опция «Видеопоток» - выводит изображение видео с

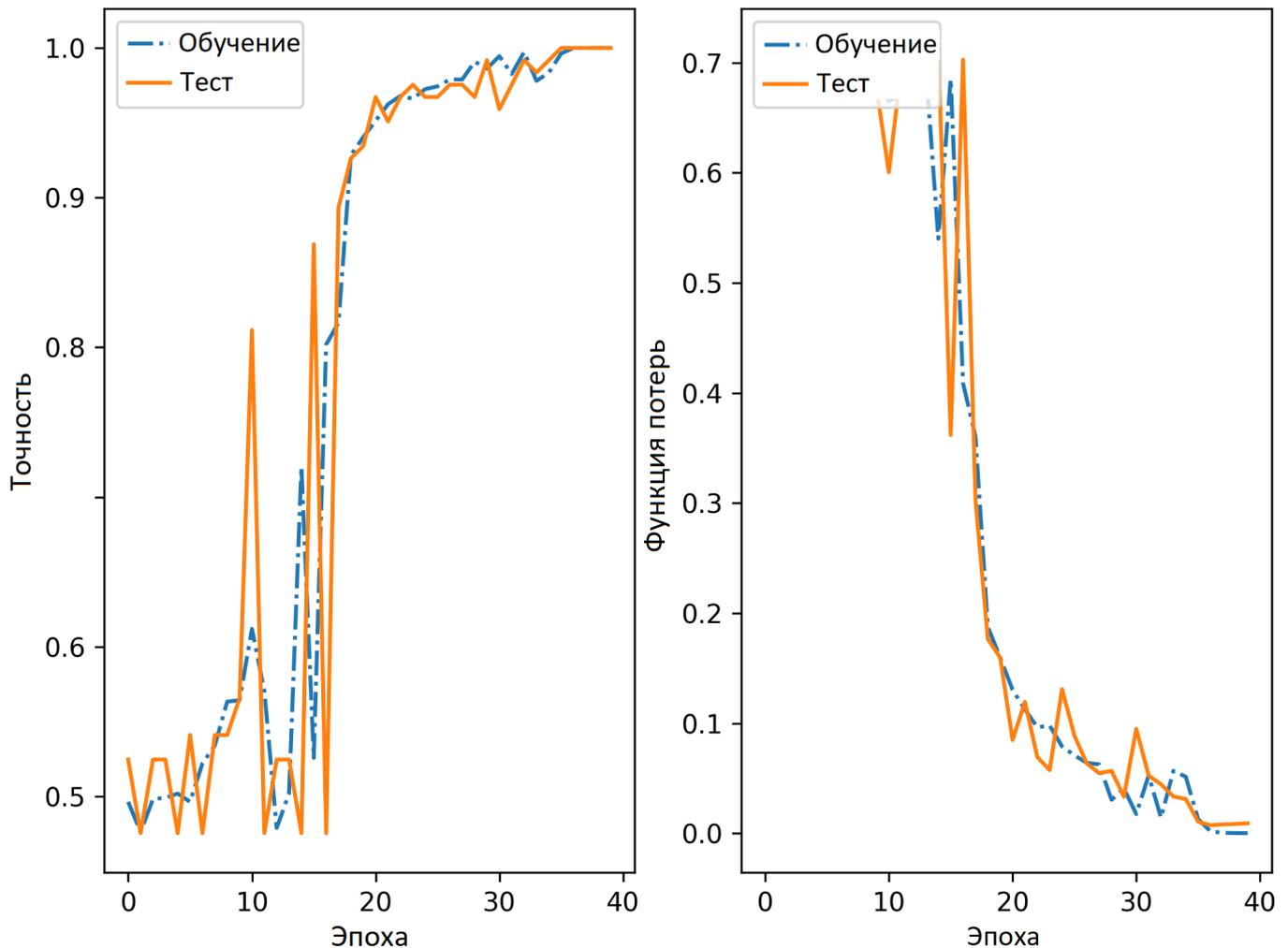


Рис. 3. График точности и функции потерь для part 5

разметкой на экран; окно для ввода названия деталей в таре при разметке; кнопка «Загрузить фото» с опциями «Из папки» и «Все файлы» - выполняет разметку выбранных фотографий из указанной папки; поле ввода расширения файлов изображений при сохранении (размеченные изображения); опция «Обратный QR» - если на QR-метках сделана инвертация цветов. Кроме того, существует возможность выбора типа нейронных сетей из списка – для детектирования ячеек (список «Тип тары») и для классификации объектов внутри ячеек (список «Тип детали»). При нажатии на кнопку «Настройка» появляется дополнительное окно настроек, позволяющее задать тип источника – файл, веб-ресурс или камера.

При нажатии на кнопку «Запуск» происходит процесс покадровой разметки (в случае видео), с использованием обученных ранее нейронных сетей. При нажатии на кнопку «Снимок» выполняется остановка процесса на текущем кадре. При нажатии на кнопку «Сохранить» выполняется сохранение кадра в ту же папку, где находится видео, к названию добавляется номер кадра.

Файл разметки сохраняется в формате JSON, содержит информацию о номере кадра (в случае видео), типе детектируемого объекта, координатах ограничивающего прямоугольника (координаты левой верхней точки, ширина и высота), а также степень уверенности в правильности классификации объекта внутри прямоугольника.

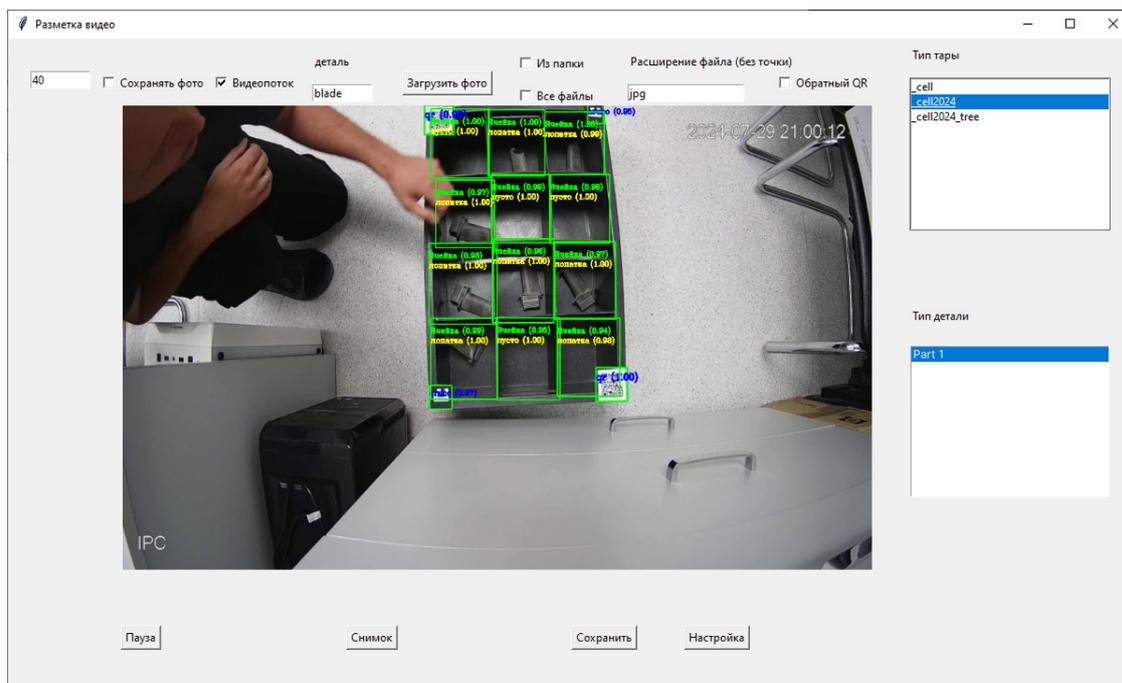


Рис. 4. Приложение во время работы

Было измерено, что в среднем трудоемкость разметки одного кадра за счет использования приложения сократилась с 82,86 с до 3,92 с.

4. ВЫВОДЫ

Разработанное приложение для разметки видеоматериала призвано сократить трудоемкость получения меток для обучения нейронных сетей и основывается на использовании уже обученных нейронных сетей, которые требуется дообучить за счет существенного увеличения выборки обучения. Использование нескольких нейронных сетей безусловно снижает скорость обработки поступающей информации – в рассматриваемых примерах почти в 4 раза – так как в нем используются последовательно три сети YOLO и одна VGG для классификации изображений внутри ячеек. Тем не менее в условиях серийного производства этот факт не критичен – скорость работы существенно выше, чем если бы разметку выполнял человек.

СПИСОК ЛИТЕРАТУРЫ

1. Zakharov O.V., Lysenko V.G., Ivanova T.N., “Asymmetric morphological filter for roughness evaluation of multifunctional surfaces,” *ISA Transactions*, vol. 146, pp. 403–420, 2024.
2. Zakharchenko M.Yu., Kochetkov A.V., Salov P.M., Zakharov O.V., “New system of functional parameters for surface texture analysis,” *Materials Today: Proceedings*, vol. 38, Part 4, pp. 1866–1870, 2021.
3. van Mourik S., van der Tol P.P.J., Linker R., Reyes-Lastiri D., Kootstra G., Koerkamp P. G., van Henten E. J., “Introductory overview: Systems and control methods for operational management support in agricultural production systems,” *Environmental Modelling & Software*, vol. 139, 105031, 2021.
4. Gulenko O., Yang H., Kim K., Youm J. Y., Kim M., Kim Y., Jung W., Yang J.-M. “Deep-Learning-Based Algorithm for the Removal of Electromagnetic Interference Noise in Photoacoustic Endoscopic Image Processing,” *Sensors*, vol. 22(10), 3961, 2022.

5. Muthukumar K. A., Bouazizi M., Ohtsuki T., “An Infrared Array Sensor-Based Approach for Activity Detection, Combining Low-Cost Technology with Advanced Deep Learning Techniques,” *Sensors*, vol. 22, no. 10, 3898, 2022.
6. Zagitov A., Chebotareva E., Toshev A., Magid E., “Comparative analysis of neural network models performance on low-power devices for a real-time object detection task,” *Computer Optics.*, vol. 48, no. 2, pp. 242-252, 2024.
7. ISO-21849-2022: “Aircraft and space — Industrial data — Product identification and traceability,” 2nd ed.; ISO: CH-1214 Vernier, Geneva, Switzerland, pp. 1-5, 2022
8. Redmon J., Divvala S., Girshick R., Farhadi F., “You only look once: Unified, real-time object detection,” 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.
9. Bhuma C. M., Kongara R., “A Novel Technique for Image Retrieval based on Concatenated Features Extracted from Big Dataset Pre-Trained CNNs” I. J. Image, Graphics and Signal Processing, vol. 2, pp. 1-12, 2023.
10. Bottou L., Bousquet O., “The Tradeoffs of Large Scale Learning,” *Advances in Neural Information Processing Systems*, NIPS Foundation, pp. 161-168, 2008
11. Kingma D., Ba J., “Adam: A Method for Stochastic Optimization,” *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, pp. 1-15, 2015.

The development of software tools for data preparation for computer vision systems in serial production

V. A. Zotov, E. Yu. Pechenina, V. A. Pechenin

An application for labelling video material has been developed. This allows, under conditions of production, limited quantity and wide nomenclature, to automation of video material labelling for further training of neural networks used in computer vision systems. To achieve sufficient accuracy with a relatively small training sample size, a cascaded identification algorithm using multiple neural networks for partitioning has been developed. During the testing phase, the $mAP_{\{50-95\}}$ metric achieved a value of 0.93.

KEYWORDS: production process management; computer vision; neural network; MDC system; object detection.