ГЕОМЕТРИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ, ЦИФРОВАЯ ПОДДЕРЖКА ЖИЗНЕННОГО ЦИКЛА ИЗДЕЛИЙ

УДК 004.42

Н. Т. СУХАНОВА, канд. пед. наук, доц. кафедры информационных систем и технологий; М. А. ЗИМИН, магистрант кафедры информационных систем и технологий

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ КВАНТОВАНИЯ ДЕТЕКТОРОВ ОБЪЕКТОВ В МОБИЛЬНЫХ СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ РЕАЛЬНОГО ВРЕМЕНИ

ФГБОУ ВО «Нижегородский государственный архитектурно-строительный университет».
Россия, 603952, г. Н. Новгород, ул. Ильинская, д. 65.
Тел.: (831) 430-54-92; эл. почта: ntsuhanova@gmail.com
Ключевые слова: техническое зрение, компьютерное зрение, детекция объектов, нейронные сети, квантование моделей, мобильные системы, реальное время, обработка изображений, инференс, препроцессинг, постпроцессинг.

В статье рассматриваются методы детекции объектов для их интеграции в мобильные системы технического зрения (СТЗ). Основное внимание уделяется моделям семейства YOLO, процедуры квантования параметров которых дают возможность адаптировать высокопроизводительные детекторы для работы на устройствах, которые имеют ограниченные вычислительные ресурсы. Представлена сравнительная оценка трех вариантов модели распознавания объектов YOLOv8n: без квантования, с частичным и с полным целочисленным квантованием. Эксперименты проводились на смартфоне Samsung Galaxy A53 5G в режиме обработки видеопотока в реальном времени. Установлено, что полное целочисленное квантование не дает ожидаемого ускорения и, напротив, снижает общую производительность за счет возрастания затрат на подготовку входных данных. Одновременно наблюдается резкое падение качества распознавания, модель перестает обнаруживать большинство объектов в кадре. Вариант с частичным квантованием показал результаты, практически идентичные базовой модели, и признан наиболее сбалансированным решением. В работе демонстрируется возможность трансформации мобильного устройства в отдельный модуль технического зрения с детерминированным временем отклика. Обосновывается для одностадийных детекторов выбор оптимальных параметров квантования, работающих в составе распределенных систем технического зрения, это, в свою очередь, позволяет минимизировать вычислительную нагрузку без потери достоверности данных. Рассмотренная методология может найти свое применение в процессе разработки мобильных систем мониторинга, промышленной инспекции и сервисной робототехники.

Введение

Современная парадигма развития систем технического зрения (СТЗ) направлена, в частности, на перенос вычислительной нагрузки на сенсорные узлы, что обуславливает актуальность разработки мобильных приложений с интегрированными алгоритмами распознавания объектов. Реализация инференса на конечном устройстве (*edge computing*) позволяет минимизировать временные

задержки при обработке видеопотока и обеспечить функциональную устойчивость системы в условиях нестабильного канала связи.

В качестве базового алгоритмического решения для подобных СТЗ эффективно применение детекторов *YOLO* (*You Only Look Once*) [1, 2]. Одноступенчатая архитектура данных нейронных сетей, которая интегрирует этапы регрессии координат ограничивающих рамок, локализацию и классификацию объектов в единый вычислительный цикл, обеспечивая тем самым оптимальный баланс между вычислительной сложностью и точностью локализации объектов. Использование методов оптимизации графа вычислений и квантования параметров моделей *YOLO* позволяет трансформировать мобильное устройство в инструмент технического зрения, способный функционировать в контурах реального времени с заданными метриками достоверности распознавания.

Однако использование на мобильных устройствах нейросетевых моделей сталкивается с трудностями, связанными с незначительными вычислительными ресурсами, которыми они располагают. При этом приходится делать выбор, связанный либо с точностью распознавания объектов, либо со скоростью вычислений [3, 4].

Одним из ключевых инструментов решения данной проблемы является квантование – перевод весовых коэффициентов и активаций нейронной сети из 32-битного представления с плавающей точкой в 16- или 8-битное. Квантование позволяет выполнять инференс на целочисленной арифметике, которая реализуется на мобильном устройстве эффективнее, чем операции с плавающей точкой, и дает дополнительный выигрыш в виде уменьшения размера модели [5].

В статье приводится сравнительная оценка трех конфигураций модели *YOLOv*8*n* при использовании в мобильном приложении на платформе *Android* в режиме реального времени. Для достижения поставленной цели решаются следующие задачи:

– экспорт исходной модели в три формата *TFLite*, соответствующие уровням точности *Float* 32, *Float* 16 и *Int*8;

– измерение производительности каждой конфигурации по следующим метрикам: среднее время инференса, частота кадров и потребления оперативной памяти;

– оценка точности детекции на наборе тестовых изображений;

– определение конфигурации, обеспечивающей оптимальный баланс между скоростью и точностью работы.

Материалы и методы

Проблема существующих исследований

Сохан М., СайРам Т., РамиРедди К. В. в обзорной работе по архитектуре *YOLOv*8 [6] показывают, что ключевым отличием этой модели стала разделенная голова детекции (*Decoupled-Head*), которая разделяет задачи классификации и регрессии. Это позволяет ветвям обучаться независимо, исключая взаимное влияние и повышая точность предсказаний. Компактная конфигурация *YOLOv*8*n* с 3,2 млн параметров и точностью 37,3 % *mAP* позиционируется авторами как оптимальная для маломощных платформ. В свою очередь, Небаба и Марков, исследуя *YOLO*-архитектуры в мобильных системах, определили порог реального времени – не менее 25 кадров в секунду (*FPS*) [3].

Джейкоб Б., Клигис С., Чен Б. и др. показали, что перевод модели в формат *Int*8 ускоряет вычисления на мобильном устройстве, однако для компактных моделей такое квантование без специальной подготовки приводит к заметной потере точности [5]. Мун Х.-Ч., Ли С., Чон Дж., Ким С. [7] установили конкретную причину этой проблемы применительно к *YOLO*-детекторам: при конвертации в *Int*8 разные части модели работают в несовместимых числовых диапазонах, из-за чего детектор перестает находить объекты, вплоть до полного отказа детекции, что авторы обосновали и доказали, в том числе, для *YOLOv*8*n*.

В исследовании Ли З., Паолиери М., Голубчик Л. [8] было доказано, что теоретическое количество вычислительных операций (*Flops*) не позволяет точно предсказать реальную скорость работы нейросети. Ошибка в таких прогнозах может достигать 43 %, что делает метрику *Flops* ненадежной. Однако данная работа фокусировалась только на задачах классификации. В то же время стандартная оценка детекторов на наборе данных *MS COCO*, описанная в статье Лин Т.-Й., Майр М., Белонжи С. и др. [9], не отражает специфику работы мобильных устройств в режиме реального времени. Таким образом, тестирование трех уровней квантования *YOLOv*8*n* непосредственно на платформе *Android* заполняет этот пробел и определяет актуальность данной статьи.

Описание модели и эксперимента

В качестве объекта исследования выбрана *YOLOv*8*n*, наиболее компактная конфигурация в семействе *YOLOv*8 [10]. Выбор данной модели обусловлен ее ориентацией на мобильные платформы при сохранении конкурентоспособной точности. Исходная модель, предобученная на наборе данных *MS COCO,* была конвертирована в формат *TensorFlow Lite* в трех вариантах: без потери точности (*Float* 32), с квантованием весов до 16 бит (*Float* 16) и с полным 8-битным целочисленным квантованием (*INT*8) методом *post-training quantization*.

Эксперименты проводились на смартфоне *Samsung Galaxy A*53 5*G*, оснащенном процессором *Exynos* 1280 (*GPU Mali-G*68) и 8 ГБ оперативной памяти, версия *Android* 15. Тестирование на единственном устройстве позволило исключить влияние аппаратных различий на итоговые показатели.

Оценка быстродействия проводилась в режиме обработки видеопотока с использованием *GPU*-делегата. Для каждой конфигурации обрабатывалась последовательность из 300 кадров; при этом первые 30 итераций исключались из расчетов для стабилизации работы графического ускорителя. Задержка измерялась на уровне системного таймера с наносекундной точностью, а потребление ресурсов оценивалось по объему динамически выделяемой нативной памяти. Целевым ориентиром производительности принято минимум 25 *FPS* [3].

Для оценки качества детекции сформирован тестовый набор из 20 изображений выборки *MS COCO* [7]. Выборка охватывает четыре типичных бытовых сценария (кухня, рабочий стол, жилая комната и улица), что позволяет проверить устойчивость модели к разным условиям освещения и плотности объектов. Объекты считались обнаруженными при пороге уверенности (*confidence threshold*) выше 0,45. Основными метриками оценки служат точность (*precision*) и полнота (*recall*), характеризующие долю верных срабатываний и способность модели находить все целевые объекты соответственно.

Анализ результатов

В данной статье представлены результаты экспериментального исследования. Сводные данные по временным затратам на различных этапах обработки кадра приведены в табл. 1.

Таблица 1

Временные характеристики и потребление памяти различными конфигурациями модели

Конфигу-рация	Инфе-ренс, мс	Препро-цессинг, мс	Постпро-цессинг, мс	Общее время, мс	*FPS*	*RAM*, Мб
*Float*32	39,39	13,53	6,91	60,64	16,49	75,3
*Float*16	40,06	13,60	6,89	61,37	16,29	75,5
*Int*8	41,96	38,50	1,69	82,74	12,09	74,1

Анализ данных табл. 1 выявил снижение производительности конфигурации *Int*8 (падение *FPS* на 36,4 % относительно *Float* 32). Детальный разбор стадий обработки показал, что основным препятствием для роста скорости стал этап препроцессинга, накладные расходы на который кратно возрастают при использовании целочисленного квантования. Это подтверждает, что теоретический выигрыш от быстрого инференса, описанный в [5], может быть полностью нивелирован затратами на подготовку входных данных. При этом разница в производительности между конфигурациями *Float* 32 и *Float* 16 является статистически незначимой.

График на рис. 1 показывает критическое доминирование этапа препроцессинга в конфигурации *Int*8. В отличие от *Float*-версий, где распределение ресурсов остается сбалансированным, в 8-битной модели время подготовки данных кратко превышает время самого инференса. Также стоит отметить, что ни одна из конфигураций не достигла порога в 25 *FPS*, необходимого для плавного восприятия видеопотока в реальном времени.

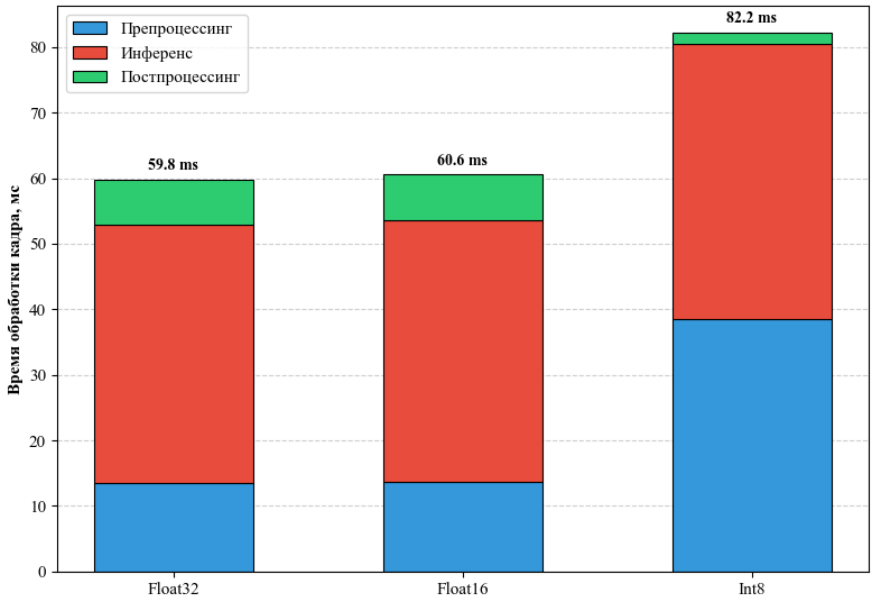

Рис. 1. Структура временных затрат на обработку одного кадра видеопотока

В табл. 2 приводятся сравнительный анализ точности детекции объектов на тестовой выборке.

Таблица 2

Сравнение точности детекции объектов на тестовой выборке

Конфигурация	*Precision*	*Recall*	*F1-score*
*Float*32	0,964	0,380	0,545
*Float*16	0,966	0,394	0,560
*Int*8	0,833	0,141	0,241

Сравнительный анализ точности в табл. 2 фиксирует стабильность метрик для формата *Float*16, однако для *Int*8 наблюдается катастрофическое падение полноты детекции (0,141). Это означает пропуск более 85 % объектов, что экспериментально подтверждает выводы Муна и др. [7] о деградации данной архитектуры при стандартном квантовании.

Представленная на рис. 2 диаграмма иллюстрирует критическую деградацию характеристик модели в формате *Int*8. При сохранении относительно высоких показателей точности значение полноты стремится к минимуму. Для обеспечения работоспособности модели в формате *Int*8 необходимо применение *quantization-aware training* [7].

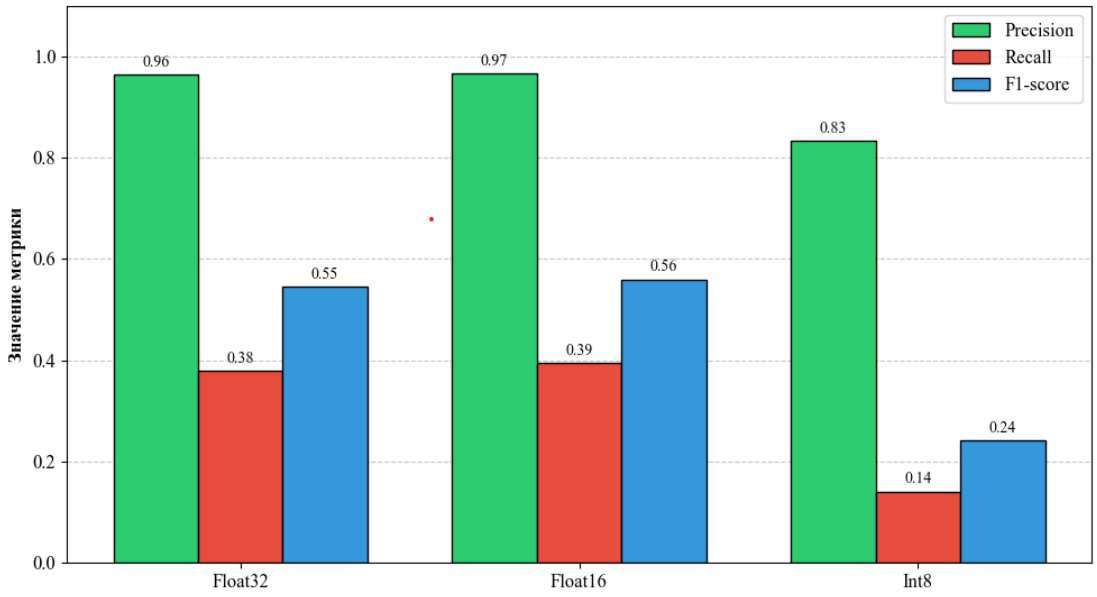

Рис. 2. Сравнение метрик точности моделей при квантовании

Заключение

Проведенное исследование методов оптимизации нейросетевых детекторов для мобильных систем технического зрения (СТЗ) позволило сформулировать следующие выводы:

Установлено, что формат *Int*8, теоретически обеспечивающий наибольшее ускорение, на практике продемонстрировал худший результат. Время подготовки входных данных возросло до 38,50 мс против 13,53 мс у модели *Float*32, из-за чего общая производительность снизилась на 26,7 %. Помимо падения скорости, полнота детекции (*Recall*) упала до 0,141, то есть модель не обнаруживает более

85 % объектов на изображении. В таком виде модель непригодна для работы в режиме реального времени. Возможным решением является применение *quantization-aware training*.

Наиболее сбалансированной конфигурацией для мобильных СТЗ признан формат *Float*16. Он сохраняет точность на уровне базовой модели, а разница в скорости с *Float*32 составляет менее 1,3 %. При этом ни одна из конфигураций не достигла порогового значения в 25 *FPS*, наилучший результат составил 16,49 *FPS* у модели *Float*32. Таким образом, для достижения плавного видеопотока необходима дополнительная оптимизация препроцессинга, в частности, перенос этого этапа на нативный код.

Научная значимость работы заключается в анализе влияния этапа предобработки данных на общую производительность мобильных систем компьютерного зрения. Выполненный подбор методов нейросетевых архитектур под ограниченные возможности мобильных устройств дополняет методологию проектирования современных СТЗ.

Перспективы дальнейших исследований связаны с разработкой методов квантования с учетом особенностей мобильных платформ и оптимизацией вычислительно затратных этапов обработки данных. Это позволит минимизировать потери, связанные с энергозатратами и скоростью обработки изображений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мурадов, Ю. Компьютерное зрение и искусственный интеллект / Ю. Мурадов, Р. Атаев, А. Беглиева // Матрица научного познания. – 2024. – № 12–2. – С. 60–63.

2. Адаптивные системы технического зрения : монография / В. И. Сырямкин, М. В. Сырямкин, Д. В. Титов [и др.]. – 2-е изд., доп. – Москва : Русайнс, 2026. – 448 с. – ISBN 978-5-466-09728-3.

3. Небаба, С. Г. Сверточные нейронные сети семейства YOLO для мобильных систем технического зрения / С. Г. Небаба, В. В. Мясников // Компьютерная оптика. – 2024. – Т. 48, № 3. – С. 414–425.

4. Мамадаев, И. М. Оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS / И. М. Мамадаев, А. М. Минитаева // Программные системы и вычислительные методы. – 2024. – № 2. – С. 86–98.

5. Quantization and training of neural networks for efficient integer-arithmetic-only inference / Jacob B. [et al.] // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2018. – С. 2704–2713.

6. Sohan, M. A review on yolov8 and its advancements / M. Sohan, Sai Ram T., C. V. Rami Reddy // International conference on data intelligence and cognitive informatics. – Singapore : Springer, 2024. – С. 529–545.

7. YOLOv6+: simple and optimized object detection model for INT8 quantized inference on mobile devices / Moon, H. C. [et al.] // Signal, image and video processing. – 2025. – Т. 19. – № 8. – С. 665.

8. Li, Z. A benchmark for ML inference latency on mobile devices / Z. Li, M. Paolieri, L. Golubchik // Proceedings of the 7th International Workshop on Edge Systems, Analytics and Networking. – 2024. – С. 31–36.

9. Microsoft COCO: common objects in context / T.-Y. Lin, M. Maire, S. Belongie [et al.] // European conference on computer vision. – Cham : Springer International Publishing, 2014. – С. 740–755.

10. Проворова, А. А. Сравнение эффективности применения различных подходов в задаче детекции объекта на изображении низкого качества / А. А. Проворова, И. Ю. Полякова, Е. В. Кузьмичева // Научная визуализация. – 2024. – Т. 16, № 3. – С. 1–13.

SUKHANOVA Nadezhda Timofeevna, candidate of pedagogical sciences, associate professor of the chair of information systems and technologies; ZIMIN Mikhail Aleksandrovich, master degree student of the chair of information systems and technologies

COMPARATIVE ANALYSIS OF QUANTIZATION METHODS FOR OBJECT DETECTORS IN REAL-TIME MOBILE VISION SYSTEMS

Nizhny Novgorod State University of Architecture and Civil Engineering.
65, Iljinskaya St., Nizhny Novgorod, 603952, Russia.
Tel.: (831) 430-54-92; e mail: ntsuhanova@gmail.com
Key words: machine vision, computer vision, object detection, neural networks, model quantization, mobile systems, real-time, image processing, inference, preprocessing, postprocessing.

This article examines object detection methods for integration into mobile machine vision systems (MVS). It focuses on models from the YOLO family, whose parameter quantization procedures enable the adaptation of high-performance detectors for devices with limited computing resources. A comparative evaluation of three variants of the YOLOv8n object recognition model is presented: without quantization, with partial quantization, and with full integer quantization. Experiments were conducted on a Samsung Galaxy A53 5G smartphone processing a real-time video stream. It was found that full integer quantization does not provide the expected speedup and, on the contrary, reduces overall performance due to increased input data preparation costs. At the same time, a sharp decline in recognition quality is observed, with the model ceasing to detect most objects in the frame. The variant with partial quantization demonstrated results virtually identical to the baseline model and is recognized as the most balanced solution. The paper demonstrates the feasibility of transforming a mobile device into a standalone machine vision module with a deterministic response time. The selection of optimal quantization parameters for single-stage detectors operating within distributed machine vision systems is substantiated. This, in turn, allows for minimizing the computational load without compromising data reliability. The methodology discussed can be applied in the development of mobile monitoring systems, industrial inspection systems, and service robotics.

REFERENCES

1. Muradov Yu., Ataev R., Beglieva A. Kompyuternoe zrenie i iskusstvenny intellekt [Computer vision and artificial intelligence]. Matritsa nauchnogo poznaniya [Matrix of Scientific Cognition]. 2024, № 12-2, P. 60–63.

2. Syryamkin V. I., Syryamkin M. V., Titov D. V. [et al.] Adaptivnye sistemy tekhnicheskogo zreniya [Adaptive machine vision systems]. Izd. 2-e, dop. Moscow, Rusains, 2026, 448 p. ISBN 978-5-466-09728-3.

3. Neibaba S. G., Myasnikov V. V. Svertochnye neyronnye seti semeystva YOLO dlya mobilnykh sistem tekhnicheskogo zreniya [Convolutional neural networks of the YOLO family for mobile machine vision systems]. Kompyuternaya optika [Computer Optics]. 2024, Vol. 48, № 3, P. 414–425.

4. Mamadaev I. M., Minitaeva A. M. Optimizatsiya proizvoditelnosti algoritmov raspoznavaniya izobrazheniy na osnove mashinnogo obucheniya dlya mobilnykh ustroystv na baze operatsionnoy sistemy iOS [Optimization of the performance of image recognition algorithms based on machine learning for mobile devices running the iOS operating system]. Programmnye sistemy i vychislitelnye metody [Software Systems and Computational Methods]. 2024, № 2, P. 86–98.

5. Jacob B. [et al.] Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, P. 2704–2713.

6. Sohan M., Sai Ram T., Rami Reddy C. V. A review on yolov8 and its advancements. International conference on data intelligence and cognitive informatics. Singapore: Springer, 2024, P. 529–545.

7. Moon H. C. [et al.] YOLOv6+: simple and optimized object detection model for INT8 quantized inference on mobile devices. Signal, image and video processing. 2025, Vol. 19, No. 8, P. 665.

8. Li Z., Paolieri M., Golubchik L. A benchmark for ML inference latency on mobile devices. Proceedings of the 7th International Workshop on Edge Systems, Analytics and Networking. 2024, P. 31–36.

9. Lin T.-Y., Maire M., Belongie S. [et al.] Microsoft COCO: common objects in context. European conference on computer vision. Cham: Springer International Publishing, 2014, P. 740–755.

10. Provorova A. A., Polyakova I. Yu., Kuzmicheva E. V. Sravnenie effektivnosti primeneniya razlichnykh podkhodov v zadache detektsii obekta na izobrazhenii nizkogo kachestva [Comparison of the effectiveness of various approaches in the task of object detection in a low-quality image]. Nauchnaya vizualizatsiya [Scientific Visualization]. 2024, Vol. 16, № 3, P. 1–13.

Получено: 24.03.2026 г.