ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 004.932 *DOI 10.55421/3034-4689_2025_28_9_89*

М. М. Ляшева, С. А. Ляшева

ОПТИМИЗАЦИЯ МОДЕЛЕЙ СЕМЕЙСТВА YOLOV8 ДЛЯ РАБОТЫ НА УСТРОЙСТВАХ С ОГРАНИЧЕННЫМИ ВЫЧИСЛИТЕЛЬНЫМИ РЕСУРСАМИ ПРИ ДЕТЕКТИРОВАНИИ СОСТОЯНИЯ ПАРКОВОЧНЫХ МЕСТ

Ключевые слова: компьютерное зрение, глубокое обучение, YOLO, квантование, детектирование объектов, оптимизация нейронных сетей.

На сегодняшний день развитие так называемых систем «Умная парковка» связано с рациональным использованием парковочного пространства. Автоматические системы мониторинга парковочных мест направлены на решение актуальной проблемы эффективного распределения времени автовладельцев. Парковочное пространство представляет собой совокупность отдельных мест – областей, предназначенных для стоянки транспортных средств. Обычно такие области размечены заранее, однако в данной статье представлен частный случай парковочного пространства без соответствующих линий разметки, поэтому задача системы мониторинга в рассматриваемом случае заключается в обнаружении парковочных мест и распознавании их состояний (занято или свободно). Задачи обнаружения и распознавания объектов можно объединить в задачу детектирования объектов. В статье рассматривается оптимизация нейронных сетей семейства YOLOv8 для эффективного выполнения задачи детектирования состояния парковочных мест на устройствах с ограниченными вычислительными ресурсами. Основное внимание уделено применению методов статического и динамического квантования, позволяющих сократить размер модели и ускорить ее работу при сохранении приемлемой точности. Представлен сравнительный анализ методов этих подходов. Эксперименты демонстрируют, что квантование весов и активаций позволяет значительно уменьшить вычислительную сложность и объем памяти, требуемые для работы моделей семейства YOLOv8 для развертывания на устройствах с ограниченными вычислительными ресурсами. Несмотря на теоретические преимущества адаптивного подхода, динамически квантованная модель показала худшие результаты по скорости обработки (1.2 FPS) в сравнении со статическим методом. Также после статической квантизации модель обеспечивает среднюю скорость обработки 2.0 FPS, (в 1.7 раза быстрее динамически квантованной версии (1.2 FPS) и в 2.2 раза быстрее исходной модели (0.9 FPS)). При этом общее время обработки видео для статически квантованной модели составило значительно меньше по сравнению с другими вариантами (665.25 секунд).

M. M. Lyasheva, S. A. Lyasheva

OPTIMIZATION OF YOLOV8 FAMILY MODELS TO WORK ON DEVICES WITH LIMITED COMPUTING RESOURCES WHEN DETECTING THE CONDITION OF PARKING SPACES

Keywords: computer vision, deep learning, YOLO, quantization, object detection, optimization of neural networks.

Today, the development of so-called "Smart Parking" systems is associated with the rational use of parking space. Automatic parking monitoring systems are aimed at solving the urgent problem of efficient allocation of car owners' time. A parking space is a collection of individual spaces – areas intended for parking vehicles. Usually such areas are marked up in advance, however, this article presents a special case of a parking space without appropriate marking lines, so the task of the monitoring system in this case is to detect parking spaces and recognize their states (occupied or vacant). The tasks of object detection and recognition can be combined into an object detection task. The article discusses the optimization of neural networks of the YOLOv8 family to effectively perform the task of detecting the state of parking spaces on devices with limited computing resources.. The main focus is on the use of static and dynamic quantization methods to reduce the size of the model and speed up its operation while maintaining acceptable accuracy. A comparative analysis of the methods of these approaches is presented. Experiments demonstrate that quantization of weights and activations can significantly reduce the computational complexity and memory required for YOLOv8 family models to be deployed on devices with limited computing resources. Despite the theoretical advantages of the adaptive approach, the dynamically quantized model showed worse results in terms of processing speed (1.2 FPS) compared to the static method. Also, after static quantization, the model provides an average processing speed of 2.0 FPS, (1.7 times faster than the dynamically quantized version (1.2 FPS) and 2.2 times faster than the original model (0.9 FPS)). At the same time, the total video processing time for the statically quantized model was significantly less than in other variants (665.25 seconds).

Введение

Современные системы детектирования объектов разрабатываются на основе технологий компьютерного зрения [1-4]. Такие технологии принято делить на методы машинного и глубокого обучения, которые чаще всего воспринимаются синонимичными терминами. Несмотря на это в данной статье методы четко ограничиваются следующим образом:

1. Методы машинного обучения, классифицирующиеся как традиционные методы, опирающиеся на предварительное формирование признаков;

2. Методы глубокого обучения, классифицирующиеся как предусматривающие автоматическое извлечение признаков.

Для достижения лучшего результата при разработке системы применяется гибридный подход, включающий в себя методы машинного и глубокого обучения. Но развертывание таких систем на устройствах с ограниченными вычислительными ресурсами, в том числе без ускорителей (GPU или CUDA), представляет собой актуальную проблему, решение которой требует высокой точности и скорости обработки данных.

Прежде чем приступать к разработке системы мониторинга необходимо определиться с выбором программного инструментария. Для обнаружения транспортных средств на парковочном месте как точной быстрый метод с корректной и устойчивой работой в условиях вариативности окружения применяются модели семейства YOLO. В рамках статьи рассматриваются версии моделей YOLOv5[5] и YOLOv8[6]. Однако стандартные реализации YOLO, даже в своих облегченных вариантах (например, YOLOv5s или YOLOv8n), могут быть избыточны для узкоспециализированной задачи определения занятости парковочных мест. Поэтому возникает необходимость в дополнительной оптимизации архитектуры нейронных сетей с учетом специфики задачи и ограничений на развертываемом устройстве.

Нейронная сеть – это математическая модель, вдохновленная биологическими нейронными сетями в головном мозге человека. Она состоит из различных компонентов, в том числе, весов и функций активации [7], которые определяют, как нейронная сеть обрабатывает данные, и они чаще всего представлены в формате с плавающей запятой (например, FP32).

Веса версий моделей YOLOv5 и YOLOv8 и размер требуемой памяти приведены в таблице ниже (см. таблицу 1) в порядке возрастания.

Таблица 1 – Веса версий моделей YOLOv5 и YOLOv8

Table 1 – Weights of model versions YOLOv5 and YOLOv8

Модель	Веса	Требуемая память (МБ)
YOLOv5n	1867405	7.12
YOLOv8n	3157216	12.04
YOLOv5s	7225885	27.56
YOLOv8s	11166560	42.6
YOLOv5m	21172173	80.77
YOLOv8m	25902688	98.81
YOLOv8l	43691520	166.67
YOLOv5l	46533693	177.51
YOLOv8x	68151392	259.98
YOLOv5x	86705005	330.75

Операции с данными в формате с плавающей точкой сильнее всего загружают процессор, так как для их хранения могут потребоваться несколько сотен мегабайт. Для решения такой проблемы существуют различные подходы, из которых можно выделить процесс квантования [8-14] для ускорения вычислений нейронной сети.

Решение

Квантизация – это процесс преобразования весов и функций активации из формата с плавающей точкой в формат с фиксированной точкой для уменьшения размера модели нейронной сети, ускорения выполнения операций и снижения объемов памяти, необходимых для хранения модели [15].

В моделях YOLO данные находятся в формате FP32, поэтому если рассматривать квантизацию преобразования из указанного формата в формат INT8, то имеет место быть следующий алгоритм:

1. Значения в формате FP32 масштабируются до диапазона INT8 одним из подходов к преобразованию значений:
   – симметричная квантизация,
   – асимметричная квантизация.

В симметричной квантизации диапазон значений симметричен относительно нуля:[−128, 127] и рассчитывается по формуле:

$$Q(x) = round\left(\frac{x}{scale}\right), \quad (1)$$

где x – исходное значение в формате FP32, scale – масштабирующий коэффициент, который вычисляется по формуле:

$$scale = \frac{\max(|x_{\min}|, |x_{\max}|)}{2^{n-1}-1}, \quad (2)$$

где n – количество бит, Q(x) – преобразованное значение в формате INT8.

В асимметричной квантизации диапазон значений не симметричен относительно нуля (например, [0, 255]) и рассчитывается по формуле:

$$Q(x) = round\left(\frac{x - zero_{point}}{scale}\right), \quad (3)$$

где x – исходное значение в формате FP32, scale – масштабирующий коэффициент, который вычисляется по формуле:

$$scale = \frac{x_{\max} - x_{\min}}{2^{n}-1}, \quad (4)$$

где n – количество бит, zero_point – смещение, $Q(x)$ – преобразованное значение в формате INT8.

2. Масштабированные значения округляются до ближайшего целого числа.

Квантизация, как и любое другое вмешательство в настройку нейронной сети, приводит к снижению точности ее работы, включая точность обнаружения объектов. В связи с этим, целесообразно применять квантизацию к большим моделям, например, YOLOv8x, чтобы за счет оптимизации производительности, ускорить обработку данных, приблизив к скорости более маленьких моделей.

Перед процедурой квантизации внутренняя структура модели нейронной сети YOLOv8x для визуализации статистических характеристик тензоров (весов и смещений) (Рис. 1) была графически представлена (Рис. 2) с помощью инструмента Netron (netron.app).

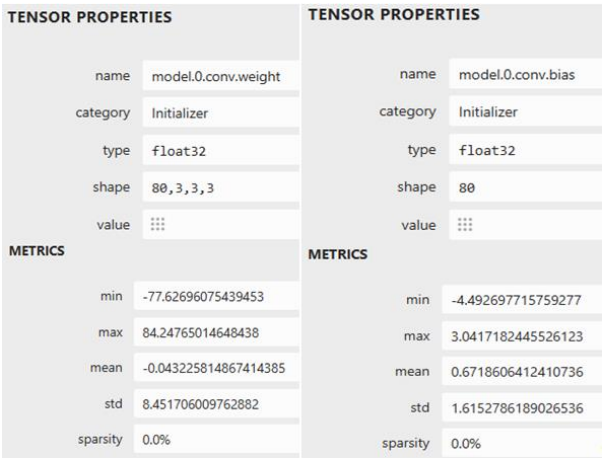

Рис. 1 – Тензоры модели YOLOv8x.onnx

Fig. 1 – Tensors of the model YOLOv8x.onnx

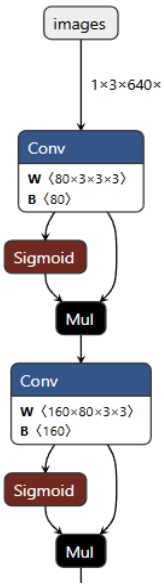

Рис. 2 – Структура модели YOLOv8x.onnx

Fig. 2 – The structure of the model YOLOv8x.onnx

Визуализация графа позволила выявить модульную структуру модели нейронной сети с повторяющимися блоками операций. Каждый блок содержит следующую последовательность: свертка (Conv), сигмоидная активация (Sigmoid) и поэлементное умножение (Mul).

Свертка (Conv, Convolution) – это операция, применяемая в нейронных сетях, которая позволяет извлекать локальные признаки из выходных данных. Для входного тензора X и ядра (фильтра) K свертка вычисляется как:

$$(X * K)_{i,j} = \sum_m \sum_n X_{i+m,j+n} \cdot K_{m,n}, \quad (5)$$

где * – оператор свертки, i, j – координаты выходного элемента.

Сигмоидная активация (Sigmoid) – это гладкая монотонно возрастающая функция, которая отображает любое вещественное число в интервал (0, 1) по формуле

$$\sigma(x) = \frac{1}{1+e^{-x}}. \quad (6)$$

Поэлементное умножение (Mul, Element-wise Multiplication) – это операция, при которой два тензора одинаковой формы перемножаются элемент за элементом по формуле:

$$(A \circ B)_{i,j} = A_{i,j} \cdot B_{i,j}, \quad (7)$$

где $\circ$ – обозначение поэлементного умножения.

Процесс квантизации разделяется на статистический и динамический. В статической квантизация веса и функция активации преобразуются заранее до работы самой модели нейронной сети, и для ее реализации требуется калибровочный набор данных. В динамической квантизации веса преобразуются заранее, а функция активации – во время выполнения модели нейронной сети.

Были исследованы оба метода сжатия моделей – статическая и динамическая квантизация. На примере модели YOLOv8x установлено, что статическая квантизация позволяет уменьшить размер модели без сильной потери точности (см. таблицу 2).

Таблица 2 – Результат сравнения исходной модели и моделей квантизации

Table 2 – The result of comparing the initial model and quantification models

Параметр	static	yolov8	dynamic
Общее время (сек)	665.25	1130.86	1441.38
Среднее время кадра (мс)	499.1 ± 20.8	848.4 ± 58.8	1081.3 ± 67.3
Средний FPS	2.0	1.2	0.9
Среднее количество объектов	0.5	0.6	0.6

Проведенные эксперименты по квантизации модели YOLOv8x показали, что статический метод обеспечивает лучшую точность и производительность по сравнению с динамическим методом. Учитывая эти результаты, далее применена статическая квантизация для моделей YOLOv8m и YOLOv8l.

Квантизированная модель YOLOv8m дает сбалансированный результат с приемлемыми потерями точности при улучшении производительности (FPS 2.61 → 4.85) (см. рис. 3).

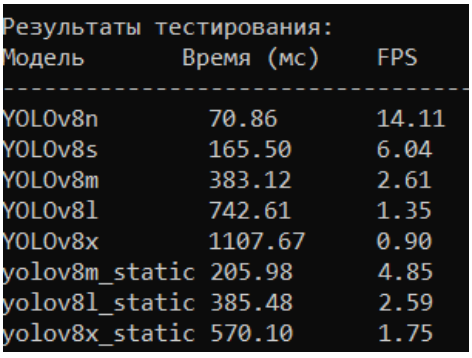

Рис. 3 – Результаты тестирования

Fig. 3 – Test results

Заключение

В ходе проведенного исследования были экспериментально оценены два метода сжатия нейронных сетей – статическая и динамическая квантизация – на примере моделей семейства YOLOv8. Основные результаты показали, что статическая квантизация демонстрирует существенные преимущества при оптимизации крупных моделей (YOLOv8x, YOLOv8l, YOLOv8m), позволяя значительно уменьшить размер модели при сохранении точности детекции для использования на устройствах с ограниченными вычислительными ресурсами.

Литература

1. Р. Дэвис, М. Терк, Компьютерное зрение. Современные методы и перспективы развития. ДМК Пресс, Москва, 2022. 690 с.
2. J. Brownlee, Deep Learning for Computer Vision: Image Classification, Object Detection and Face Recognition in Python. 2020. 563 p.
3. J. Brownlee, Machine Learning Mastery. Режим доступа в интернет: https://machinelearningmastery.com/object-recognition-with-deep-learning / Дата обращения: 06.07.25.
4. Распознавание объектов: 3 вещи, которые необходимо знать. Режим доступа в интернет: https://hub.exponenta.ru/post/raspoznavanie-obektov-3-veshchi-kotorye-neobkhodimo-znat244 / Дата обращения: 06.07.25.
5. Ultralytics YOLOv5. Режим доступа в интернет: https://docs.ultralytics.com/ru/models/yolov5 / Дата обращения: 06.07.25.
6. Ultralytics YOLOv8. Режим доступа в интернет: https://docs.ultralytics.com/ru/models/yolov8 / Дата обращения: 06.07.25.
7. R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Springer. 2022. 947 p.
8. B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2704–2713 (2018).
9. A. Goncharenko, A. Denisov, S. Alyamkin, E. Terentev, *International Journal of Computer and Information Engineering*, **13**, 9, 491–495 (2019).
10. How to Optimize a Deep Learning Model for faster Inference. Режим доступа в интернет: https://www.thinkautonomous.ai/blog/deep-learning-optimization/#how-to-optimize-a-model-for-better-performance / Дата обращения: 06.07.25.
11. Model Optimization Techniques in Neural Network: A Comprehensive Guide. Режим доступа в интернет: https://medium.com/@juanc.olamendy/model-optimization-techniques-in-neural-network-a-comprehensive-guide-322e8e88fd31 / Дата обращения: 06.07.25.
12. D. D. Lin, S. S. Talathi, V. S. *Annapureddy, Fixed point quantization of deep convolutional networks*, **6,** (2016).
13. D. Miyashita, E. Lee, B. Murmann, *Convolutional neural networks using logarithmic data representation*. arXiv preprint, arXiv:1603.01025 (2016).
14. E. Park, J. Ahn, S. Yoo, *Publisher Institute of Electrical and Electronics Engineers Inc*., **2017-January**, 7197-7205 (2017).
15. Inference optimization techniques and solutions. Режим доступа в интернет: https://nebius.com/blog/posts/inference-optimization-techniques-solutions / Дата обращения: 06.07.25.

References

1. R. Davis, M. Turk, *Computer Vision: Modern Methods and Development Prospects*. DMK Press, Moscow, 2022. 690 p.
2. J. Brownlee, Deep Learning for Computer Vision: Image Classification, Object Detection and Face Recognition in Python. 2020. 563 p.
3. J. Brownlee, Machine Learning Mastery. Internet access mode: https://machinelearningmastery.com/object-recognition-with-deep-learning / Date of access: 06.07.25.
4. Object Recognition: 3 Things You Need to Know . Internet access mode: https://hub.exponenta.ru/post/raspoznavanie-obektov-3-veshchi-kotorye-neobkhodimo-znat244 / Date of access: 06.07.25.
5. Ultralytics YOLOv5. Internet access mode: https://docs.ultralytics.com/ru/models/yolov5 / Date of access: 06.07.25.
6. Ultralytics YOLOv8. Internet access mode: https://docs.ultralytics.com/ru/models/yolov8 / Date of access: 06.07.25.
7. R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Springer. 2022. 947 p.
8. B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2704–2713 (2018).
9. A. Goncharenko, A. Denisov, S. Alyamkin, E. Terentev, *International Journal of Computer and Information Engineering*, **13**, 9, 491–495 (2019).
10. How to Optimize a Deep Learning Model for faster Inference. Internet access mode: https://www.thinkautonomous.ai/blog/deep-learning-optimization/#how-to-optimize-a-model-for-better-performance / Date of access: 06.07.25.
11. Model Optimization Techniques in Neural Network: A Comprehensive Guide. Internet access mode: https://medium.com/@juanc.olamendy/model-optimization-techniques-in-neural-network-a-comprehensive-guide-322e8e88fd31 / Date of access: 06.07.25.
12. D. D. Lin, S. S. Talathi, V. S. *Annapureddy, Fixed point quantization of deep convolutional networks*, **6,** (2016).
13. D. Miyashita, E. Lee, B. Murmann, *Convolutional neural networks using logarithmic data representation*. arXiv preprint, arXiv:1603.01025 (2016).
14. E. Park, J. Ahn, S. Yoo, *Publisher Institute of Electrical and Electronics Engineers Inc*., **2017-January**, 7197-7205 (2017).
15. Inference optimization techniques and solutions. Internet access mode: https://nebius.com/blog/posts/inference-optimization-techniques-solutions / Date of access: 06.07.25.

© М. М. Ляшева – ассистент кафедры автоматизированных систем обработки информации и управления, ФГБОУ ВО «Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ», Казань, Россия, mssmaya@mail.ru; **С. А. Ляшева** – канд. техн. наук, доцент кафедры прикладной математики и информатики, ФГБОУ ВО «Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ», Казань, Россия.

© M. M. Lyasheva – Assistant of Department for Automated Systems for Information Processing and Control, Kazan National Research Technical University named after A. N. Tupolev - KAI, Kazan, Russia, mssmaya@mail.ru; **S. A. Lyasheva** – PhD (Technical Sci.), Associate Professor of the Department of Applied Mathematics and Informatics, Kazan National Research Technical University named after A. N. Tupolev - KAI, Kazan, Russia.

Дата поступления рукописи в редакцию – 05.07.25.
Дата принятия рукописи в печать – 13.08.25.