УДК 629.7.05:62-503.5 DOI 10.46973/0201–727X_2025_2_35

*А. В. Сацюк*

# ОПТИМИЗАЦИЯ АРХИТЕКТУРЫ *YOLOv8* ДЛЯ ЗАДАЧ ЗАХВАТА ОБЪЕКТА БПЛА: АНАЛИЗ КОМПРОМИССА МЕЖДУ ТОЧНОСТЬЮ, СКОРОСТЬЮ И ВЫЧИСЛИТЕЛЬНЫМИ РЕСУРСАМИ

**Аннотация.** В работе представлен комплексный подход к оптимизации модели *YOLOv8n* для задач обнаружения объектов с использованием беспилотных летательных аппаратов в условиях ограниченных вычислительных ресурсов. Основное внимание уделено методам квантования (INT8/INT4) и прунинга (50/75%), направленным на снижение вычислительной сложности модели при сохранении приемлемой точности. В результате оптимизации разработана модель *YOLOv8n-Optimized-Drone*, демонстрирующая 4-кратный прирост скорости обработки на платформе *Raspberry Pi 5* по сравнению с базовой версией. Размер модели сокращен в 3.8 раза, что критически важно для встраиваемых систем БПЛА.

Для обучения и валидации модели создан специализированный датасет с разметкой *bounding box*, учитывающий условия съемки с БПЛА. Натурные испытания подтвердили эффективность предложенного метода, обеспечивающего баланс между производительностью, энергопотреблением и точностью. Дополнительно исследовано влияние различных уровней квантизации и прунинга на итоговые метрики, что позволило определить оптимальную конфигурацию для развертывания на маломощных устройствах. Полученные результаты открывают перспективы для дальнейшей адаптации модели к динамическим условиям полета и интеграции с мультисенсорными системами БПЛА.

**Ключевые слова:** беспилотный летательный аппарат, квантование, прунинг, оптимизация нейросетевой модели, нейронная сеть, *YOLO*, метрики *YOLO*.

**Для цитирования:** Сацюк, А. В. Оптимизация архитектуры *YOLOv8* для задач захвата объекта БПЛА: анализ компромисса между точностью, скоростью и вычислительными ресурсами / А. В. Сацюк // Вестник Ростовского государственного университета путей сообщения. – 2025. – № 2. – С. 35–42. – DOI 10.46973/0201–727X_2025_2_35.

## *Введение*

Беспилотные летательные аппараты (БПЛА) все чаще используются в решении широкого спектра задач, включая наблюдение, доставку, поиск и спасение, а также мониторинг окружающей среды. Ключевым элементом эффективной работы БПЛА является возможность автономного захвата и сопровождения объектов в режиме реального времени. Для реализации этой функциональности необходимы высокопроизводительные и энергоэффективные системы компьютерного зрения.

Одним из наиболее перспективных подходов к решению данной задачи является использование алгоритмов глубокого обучения, в частности, моделей класса *YOLO* (*You Only Look Once*). *YOLOv8*, как одна из последних итераций этой архитектуры, демонстрирует высокие результаты в задачах обнаружения объектов, обеспечивая высокую точность и скорость обработки изображений. Однако, интеграция *YOLOv8* на борту БПЛА представляет собой сложную задачу из-за ограничений по вычислительным ресурсам и энергопотреблению.

БПЛА, оснащенные микрокомпьютерами и маломощными камерами типа *MIPI*, обладают ограниченной вычислительной мощностью и объемом памяти. В таких условиях прямое применение стандартных версий *YOLOv8* может привести к неприемлемо низкой частоте кадров, недостаточной для обеспечения надежного захвата и сопровождения объекта в динамичных условиях полета.

Таким образом, возникает необходимость в оптимизации архитектуры *YOLOv8* для эффективной работы на борту БПЛА с ограниченными ресурсами.

Анализ научных публикаций показывает, что современные исследования в области обнаружения объектов с помощью БПЛА активно сосредоточены на оптимизации алгоритмов *YOLO* [1–4] для обеспечения высокой производительности, снижения энергопотребления и адаптации к ограниченным вычислительным ресурсам встраиваемых платформ. Большое внимание уделяется следующим направлениям: модификация архитектуры *YOLO*, включая создание облегченных версий и оптимизацию су-

ществующих архитектур для снижения вычислительных затрат при сохранении точности; использование трансферного обучения с применением предобученных моделей *YOLO* для ускорения обучения и повышения обобщающей способности, особенно при работе с ограниченными наборами данных, характерными для задач обнаружения объектов на БПЛА. Актуальным остается вопрос оптимизации для конкретных платформ, включая адаптацию алгоритмов *YOLO* к архитектуре процессоров, используемых в БПЛА, и к специализированным аппаратным ускорителям. Важным направлением является повышение устойчивости моделей к изменяющимся условиям освещения и погодным явлениям [5]. В ряде исследований [6–9] проводится сравнительный анализ различных алгоритмов обнаружения объектов (включая *YOLO*) на реальных БПЛА с оценкой скорости обработки, точности и энергопотребления, что подчеркивает практическую значимость данной тематики. В рассмотренных работах также все больше внимания уделяется вопросам оптимизации модели с учетом энергопотребления, что особенно актуально для БПЛА с ограниченным временем полета. Вместе с тем комплексная оптимизация, сочетающая квантование, прунинг и адаптацию к конкретным условиям эксплуатации БПЛА (включая учет динамического изменения вычислительной нагрузки и агрессивную аугментацию данных для повышения устойчивости), оставалась недостаточно изученной. В представленной работе предложен подход, направленный на восполнение этого пробела, и продемонстрирована его эффективность в реальных условиях эксплуатации БПЛА.

Целью данной работы является исследование компромисса между точностью обнаружения, скоростью обработки и потреблением вычислительных ресурсов при адаптации *YOLOv8* для задачи захвата объекта БПЛА. Результаты данного исследования позволят определить оптимальную архитектуру *YOLOv8* для автономного захвата и сопровождения объектов с использованием БПЛА, обеспечивая баланс между точностью, скоростью и энергоэффективностью.

***Основная часть***

В рамках данного исследования проведена оценка эффективности различных методов оптимизации архитектуры *YOLOv8* для задач захвата объекта беспилотным летательным аппаратом. Исследование включало в себя несколько этапов: формирование специализированного датасета, обучение базовой модели нейронной сети, реализация методов оптимизации, оценка производительности оптимизированных моделей.

Для обучения и последующей оценки производительности модели разработан специализированный датасет. Он состоит из 160 изображений целевого объекта, полученных с камеры БПЛА в различных условиях освещения и с разных углов обзора. Важным ограничением является наличие в кадре только одного экземпляра целевого объекта, что соответствует сценарию захвата и сопровождения единственной цели [10, 11]. Датасет разделен на обучающую (80 %) и валидационную (20 %) выборки. Разметка изображений выполнена посредством ограничивающих прямоугольников (*bounding boxes*), определяющих положение и размеры целевого объекта в кадре.

В качестве контрольной точки обучена базовая модель *YOLOv8n* с применением стандартных гиперпараметров, рекомендованных разработчиками: размер выходного изображения 640×640, размера батча 8, скорость обучения динамическая и установлена в приделах 0.00001-0.1. Процесс обучения осуществлялся на вычислительном кластере с использованием графических процессоров *NVIDIA GeForce GTX 1650*. Мониторинг обучения производился с использованием стандартных метрик *box_loss, cls_loss, dfl_loss, Precision, Recall, mAP50* и *mAP50-95*.

*box_loss* – функция потерь, оценивающая точность предсказанных ограничивающих прямоугольников вокруг объектов.

*cls_loss* – функция потерь, оценивающая точность классификации объектов на общем фоне.

*dfl_loss* – функция потерь, используемая в *YOLOv8* для улучшения предсказания *bounding box*. Данная функция фокусируется на обучении модели более точно предсказывать распределение вероятностей положения границ объекта, особенно в сложных случаях (например, при перекрытии объектов или при плохом освещении).

*P* (*Precision* с англ. точность) – функция, показывающая долю правильно обнаруженных объектов среди всех объектов, предсказанных моделью как целевые. Функция характеризует точность модели.

*R* (*Recall* с англ. полнота) – функция, показывающая долю правильно обнаруженных объектов среди всех фактических целевых объектов в датасете.

*mAP50* (*Mean Average Precision at IoU=0.5*) – среднее значение точности для всех классов, вычисленное при пороге равном 0.5.

*mAP50-95* (*Mean Average Precision at IoU from 0.5 to 0.95*) – среднее значение AP, вычисленное для различных порогов IoU, от 0.5 до 0.95 с шагом 0.05.

Анализ данных (рис. 1) показал, что после 60 эпох наблюдается стабилизация метрик, дальнейшее увеличение числа эпох не приводит к существенному улучшению результатов обучения.

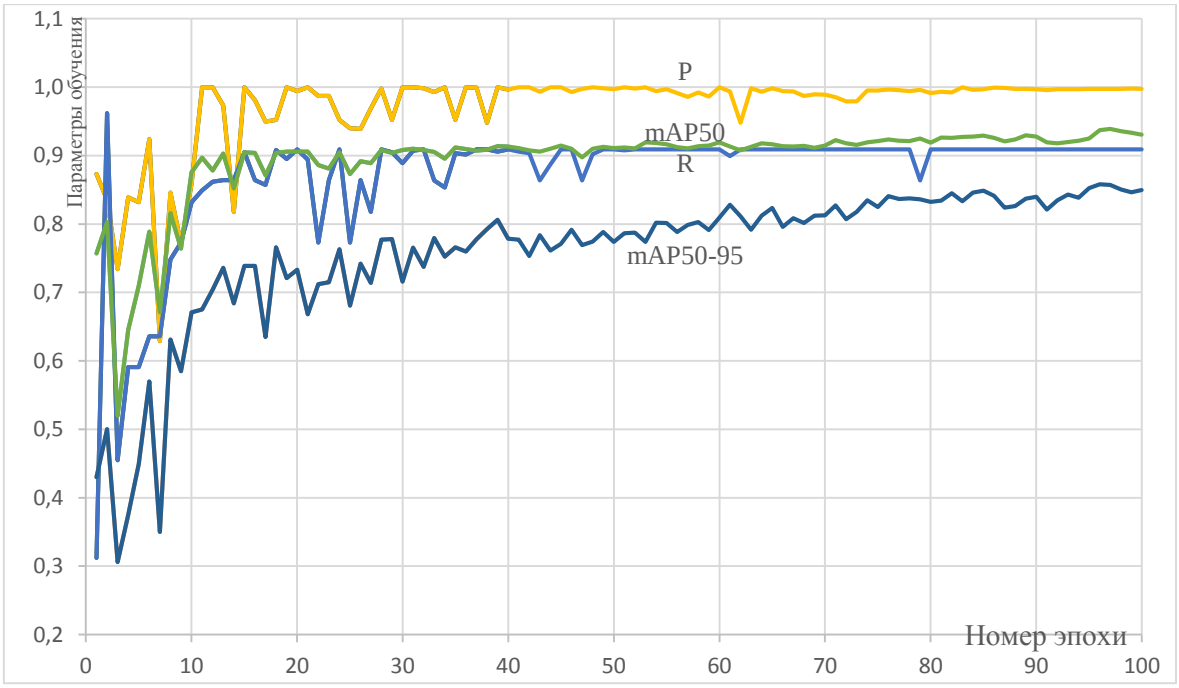


**Рис. 1. Процесс обучение нейронной сети *YOLOv8n***

В результате обучения модели *YOLOv8n* на собственном наборе данных, была получена модель *YOLOv8n-Drone*.

В рамках данного исследования, для адаптации *YOLOv8n-Drone* к условиям ограниченных вычислительных ресурсов БПЛА, были реализованы следующие методы оптимизации, нацеленные на снижение вычислительной сложности модели без существенной потери точности обнаружения:

1 *Квантизация*. Квантизация заключается в преобразовании весов и активаций нейронной сети из формата с плавающей точкой (обычно *FP32* или *FP16*) в формат с фиксированной точкой (например, *INT8* или *INT4*). Это позволяет значительно уменьшить объем памяти, необходимый для хранения модели, и ускорить вычисления, поскольку целочисленные операции выполняются быстрее и энергоэффективнее на большинстве аппаратных платформ, используемых во встраиваемых системах (микрокомпьютеры).

В данном исследовании использовалась техника *Quantization Aware Training* (*QAT*), реализованная с помощью библиотеки *PyTorch. QAT* подразумевает обучение модели с учетом эффектов квантизации. В процессе обучения в модель добавляются операции, имитирующие квантизацию и деквантизацию, что позволяет сети адаптироваться к ограниченному диапазону значений и минимизировать потери точности.

При *INT8* квантизации веса и активации квантовались до 8-битного целочисленного формата. Это обеспечивает хороший компромисс между уменьшением размера модели и сохранением точности.

При *INT4* квантизации веса и активации квантовались до 4-битного целочисленного формата. Это позволило добиться еще большего уменьшения размера модели и увеличения скорости вычислений, однако потребовало более тщательной настройки параметров обучения для компенсации потери точности.

Таким образом, в данном исследовании квантизация *YOLOv8n-Drone* позволила преобразовать веса и активации модели в целочисленный формат (*INT8* и *INT4*), значительно снизив размер модели и увеличив скорость обработки на микрокомпьютере. При этом, использование *INT4* квантизации обеспечило наибольший прирост *FPS*, но сопровождалось незначительным снижением точности обнаружения объектов. Полученные квантованные модели демонстрируют *пригодность YOLOv8n-Drone* для эффективного развертывания на встраиваемых платформах с ограниченными вычислительными ресурсами.

2 *Прунинг*. Прунинг (англ. *pruning* – «обрезка») – метод сжатия нейронной сети, направленный на уменьшение её вычислительной сложности и объёма памяти за счёт удаления избыточных или малозначимых параметров (весов, нейронов или целых слоёв). В данной работе применялся *weight pruning* – поэтапное обнуление весов с наименьшими абсолютными значениями с последующей переподготовкой модели для сохранения точности. Этот подход позволяет сократить размер модели и ускорить её работу на встраиваемых устройствах без существенной потери качества предсказаний.

В рамках исследования по оптимизации *YOLOv8n-Drone* для задач БПЛА, прунинг был применен как метод уменьшения вычислительной сложности модели путем удаления избыточных или малозначимых соединений.

В проекте использован метод *weight pruning*, заключающийся в обнулении весов с наименьшей абсолютной величиной. Этот метод прост в реализации и достаточно эффективен для уменьшения размера модели. Для упрощения процесса прунинга и управления им применялась библиотека *SparseML. SparseML* предоставляет инструменты для применения различных техник прунинга, а также для оценки и переподготовки модели после прунинга.

Прунинг не применялся однократно, а выполнялся итеративно. После каждой итерации обнуления весов, модель подвергалась переподготовке на обучающем датасете. Это позволяло восстановить часть потерянной точности и компенсировать негативное влияние прунинга.

Ключевым параметром прунинга является уровень разреженности, определяющий долю весов, которые будут обнулены. В исследовании были протестированы два уровня разреженности: 50 и 75 %.

Обнулению подвергались 50 % весов с наименьшей абсолютной величиной. Это обеспечивало умеренное уменьшение размера модели и увеличение скорости, при относительно небольшом снижении точности. Также обнулению подвергались 75 % весов с наименьшей абсолютной величиной. Это приводило к более значительному уменьшению размера модели и увеличению скорости, но требовало более тщательной переподготовки для сохранения приемлемого уровня точности.

После каждой итерации прунинга и переподготовки, производилась оценка производительности модели на валидационном датасете. Измерялись метрики *Precision, Recall, mAP50, mAP50-95*, а также *FPS* и энергопотребление на платформе *Raspberry Pi 5*.

Реализация прунинга в данном исследовании позволила значительно уменьшить размер *YOLOv8n* и повысить скорость инференса на *Raspberry Pi 5*, за счет обнуления наименее значимых весов. Увеличение уровня разреженности приводило к большему уменьшению размера модели и увеличению скорости, но требовало более тщательной переподготовки для минимизации потерь точности. Таким образом, прунинг является эффективным методом оптимизации, требующим балансировки между уровнем разреженности и усилием, затраченным на переподготовку, для достижения оптимального компромисса между размером модели, скоростью и точностью в задачах захвата объектов БПЛА.

В таблице представлены результаты экспериментов по оптимизации модели *YOLOv8n-Drone* для развертывания на встраиваемой платформе *Raspberry Pi 5*. Рассматривались различные методы оптимизации, включая квантование (*INT8* и *INT4*) и прунинг (50 и 75 %). Основными критериями оценки являлись размер модели, точность (*Precision, Recall, mAP50, mAP50-95*), скорость обработки (*FPS* и задержка) и энергопотребление.

Наиболее значимые улучшения были достигнуты при использовании комбинации квантования *INT8* и прунинга 50 % (*YOLOv8n-Optimized-Drone*). Размер модели был уменьшен до 1.56 MB, что в ~3.8 раза меньше, чем у базовой модели (5.94 MB). При этом наблюдалось лишь незначительное снижение точности*: Precision* уменьшился с 0.94 до 0.91, *Recall* – с 0.89 до 0.87, *mAP50* – с 0.92 до 0.90, а *mAP50-95* – с 0.82 до 0.80. Наиболее важным является то, что *FPS* увеличился с 6.6 до 26.4, а задержка снизилась с 152 до 38 мс. Энергопотребление также уменьшилось с 2.5 до 2.0 W.

Квантование *INT8* показало прирост *FPS* в 3 раза. Прунинг 50 % самостоятельно – в 1,5 раза, а совместно эти методы показали – в 4 раза, поэтому использование этих методов, а также в комбинации, как это демонстрируется в *YOLOv8n-Optimized-Drone*, дает выигрыш в *FPS* и задержке. Другие методы оптимизации, такие как квантование *INT4* и прунинг 75 %, привели к еще большему уменьшению размера модели, но за счет более значительного снижения точности.

**Таблица результатов экспериментов по оптимизации модели *YOLOv8n-Drone***

| Метод оптимизации | Размер модели (MB) | Precision | Recall | mAP50 | mAP50-95 | FPS (RPi 5) | Задержка (мс) | Энергопотреб. (W) |
|---|---|---|---|---|---|---|---|---|
| Базовая модель (*YOLOv8n-Drone*, 60 эпох) | 5.94 | 0.94 (0.91–0.98) | 0.89 (0.88–0.90) | 0.92 (0.91–0.93) | 0.82 (0.81–0.83) | 6.6 | 152 (146–156) | 2.5 (2.4–2.6) |
| *YOLOv8n-Optimized-Drone* (INT8 + Prune 50 %) | 1.56 | 0.91 (0.89–0.93) | 0.87 (0.85–0.89) | 0.90 (0.88–0.92) | 0.80 (0.78–0.82) | 26.4 | 38 (37–40) | 2.0 (1.9–2.1) |
| Квантизация (INT8) | 3.12 | 0.92 (0.90–0.94) | 0.88 (0.86–0.90) | 0.91 (0.89–0.93) | 0.81 (0.79–0.83) | 19.8 | 76 (73–78) | 2.1 (2.0–2.2) |
| Квантизация (INT4) | 1.56 | 0.89 (0.87–0.91) | 0.85 (0.83–0.87) | 0.88 (0.86–0.90) | 0.79 (0.77–0.81) | 9.8 | 102 (99–105) | 1.8 (1.7–1.9) |
| Прунинг (50 %) | 3.74 | 0.93 (0.91–0.95) | 0.88 (0.86–0.90) | 0.91 (0.89–0.93) | 0.81 (0.79–0.83) | 9.9 | 101 (96–108) | 2.3 (2.2–2.4) |
| Прунинг (75 %) | 1.56 | 0.88 (0.86–0.90) | 0.86 (0.84–0.88) | 0.87 (0.85–0.89) | 0.78 (0.76–0.80) | 9.1 | 110 (106–114) | 1.9 (1.8–2.0) |

Снижение точности при использовании *YOLOv8n-Optimized-Drone* (*INT8 + Prune* 50 %) является незначительным (менее 3 %) и может быть приемлемым в зависимости от конкретного приложения. Для большинства задач обнаружения объектов, где важна скорость обработки в реальном времени, небольшое снижение точности вполне оправдано значительным увеличением *FPS* и уменьшением задержки (рис. 2). Например, в системах видеонаблюдения, где требуется быстрое обнаружение потенциальных угроз, или в системах управления дронами, где важна оперативная реакция, скорость обработки имеет приоритет над максимальной точностью.

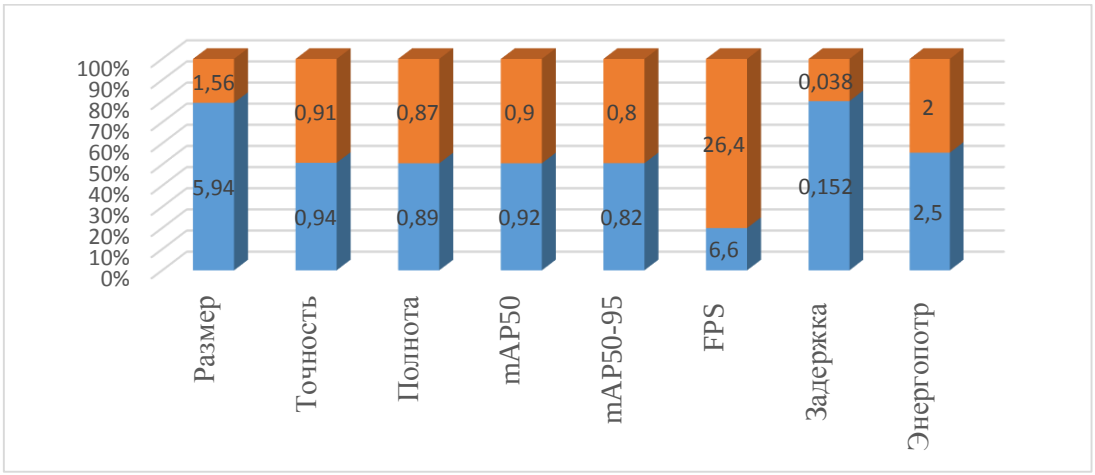


**Рис. 2. Влияние оптимизации модели *YOLOv8n-Optimized-Drone* на производительность**

В то же время дальнейшее уменьшение размера модели за счет использования квантования *INT4* или прунинга 75 % приводит к чрезмерному снижению точности, что может быть неприемлемым для многих приложений. Поэтому, учитывая достигнутый баланс между размером модели, точностью

и скоростью обработки, *YOLOv8n-Optimized-Drone* (*INT8 + Prune* 50 %) представляется оптимальным выбором для развертывания на *Raspberry Pi 5* в задачах обнаружения объектов в реальном времени. Дальнейшее увеличение точности потребовало бы использования более тяжелых моделей, что привело бы к неприемлемому снижению производительности на встраиваемой платформе.

***Выводы***

Данная работа представила систематическое исследование методов оптимизации модели *YOLOv8n* для развертывания на встраиваемой платформе *Raspberry Pi 5*, ориентированное на обнаружение объектов с применением беспилотных летательных аппаратов. Результаты, полученные в ходе натурных испытаний с использованием БПЛА в реальных условиях, подтвердили эффективность предложенного подхода. Достигнут существенный прогресс в оптимизации модели, о чем свидетельствует модель *YOLOv8n-Optimized-Drone* (*INT8 + Prune* 50 %), которая обеспечивает значительное снижение размера модели и задержки при незначительной потере точности, что делает ее оптимальной для задач обнаружения в реальном времени в условиях ограниченных ресурсов БПЛА. Дальнейшее развитие предусматривает изучение адаптивных методов оптимизации, учитывающих динамические изменения вычислительной нагрузки и внешних условий в ходе полета БПЛА, а также исследование более агрессивных методов аугментации данных для повышения устойчивости модели к вариативности условий съемки и улучшения обобщающей способности. Кроме того, планируется изучение возможности интеграции модели с другими сенсорами БПЛА для повышения точности и надежности обнаружения.

**Список литературы**

1 Улучшенный алгоритм на основе *YOLOv5* для обнаружения амброзии полыннолистной на изображениях, полученных с помощью БПЛА / С. Сяомин, С. Тяньцзэн, М. Хаомин [и др.] // Front. Plant Sci. – 2021. – Т. 15. – С. 1360419. – DOI 10.3389/fpls.2024.1360419.

2 Real-time Object Detection for UAV Aerial Images Based on Improved *YOLOv7* / Y. Zhang, X. Chen, H. Li, Y. Wang // Journal of Electronic Imaging. – 2015. – Vol. 32, No. 6. – P. 063002. – DOI 10.3390/electronics12234886.

3 **Сацюк, А. В.** Оценка эффективности алгоритмов *YOLO* для обнаружения объектов в реальном времени во встраиваемых системах беспилотных транспортных средств / А. В. Сацюк, Р. В. Белый, А. Е. Ищенко // Сборник научных трудов Донецкого института железнодорожного транспорта. – 2024. – № 4 (75). – С. 73–82. – ISSN 1993-5579.

4 **Redmon, J. A.** *YOLO9000* : Better, Faster, Stronger / J. Redmon, A. Farhadi // Conference on Computer Vision and Pattern Recognition (CVPR). – 2017. – arXiv :1612.08242v1.

5 **Сацюк, А. В.** Анализ трекеров алгоритмов компьютерного зрения в вопросах отслеживания подвижных объектов в видеопотоке / А. В. Сацюк, Е. Г. Воевода // Сборник научных трудов Донецкого института железнодорожного транспорта. – 2023. – № 71. – С. 54–65. – ISSN 1993-5579.

6 Библиотека программиста. *JavaScript* для глубокого обучения / Ф. Шолле, Э. Нильсон, С. Бэйлесчи [и др.]. – Санкт-Петербург : Питер, 2021. – 576 с. – ISBN 978-5-4461-1697-3.

**References**

1 An improved *YOLOv5-based* algorithm for common ragweed detection in UAV images / S. Xiaoming, S. Tianzeng, M. Haoming [et al.] // Front. Plant Sci. – 2021. – Vol. 15. – P. 1360419. – DOI 10.3389/fpls.2024.1360419.

2 Real-time Object Detection for UAV Aerial Images Based on Improved *YOLOv7* / Y. Zhang, X. Chen, H. Li, Y. Wang // Journal of Electronic Imaging. – 2015. – Vol. 32, No. 6. – P. 063002. – DOI 10.3390/electronics12234886.

3 **Satsuk, A. V.** Evaluation of the effectiveness of *YOLO* algorithms for real-time object detection in embedded systems of unmanned vehicles / A. V. Satsuk, R. V. Bely, A. E. Ishchenko // Collection of scientific papers of the Donetsk Institute of Railway Transport. – 2024. – No. 4 (75). – P. 73–82. – ISSN 1993-5579.

4 **Redmon, J. A.** *YOLO9000* : Better, Faster, Stronger / J. Redmon, A. Farhadi // Conference on Computer Vision and Pattern Recognition (CVPR). – 2017. – arXiv :1612.08242v1.

5 **Satsuk, A. V.** Analysis of computer vision algorithm trackers in matters of tracking moving objects in a video stream / A. V. Satsuk, E. G. Voevoda // Collection of scientific papers of the Donetsk Institute of Railway Transport. – 2023. – No. 71. – P. 54–65. – ISSN 1993-5579.

6 Programmer's Library. *JavaScript* for deep learning / F. Schollet, E. Nilsson, S. Bayleschi [et al.]. – Saint-Petersburg : Piter, 2021. – 576 p. – ISBN 978-5-4461-1697-3.

7 *YOLOv8*. Ultralytics. Официальный репозиторий *YOLOv8* / G. Jocher, A. Stoken, A. Chaly [et al.]. – 2023. – URL: https://github.com/ultralytics/ultralytics (date of access: 02/07/2025).
8 Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation / Z. Zheng, P. Wang, D. Ren [et al.] // Transactions on Cybernetics. – DOI 10.1109/TCYB.2021.3095305.
9 An Efficient UAV-Based Aerial Image Object Detection Method via Improved YOLOv4 / B. Liu, X. Li, G. Wang [et al.] // Remote Sensing. – 2023. – Vol. 12, No. 21. – P. 3599. – DOI 10.1109/SIU53274.2021.9478027.
10 **Сацюк, А. В.** Особенности разработки беспилотных летательных аппаратов для отрасли железнодорожного транспорта / А. В. Сацюк, А. А. Воробьев // Сборник научных трудов Донецкого института железнодорожного транспорта. – 2023. – № 68. – С. 13–21. – ISSN 1993-5579.
11 **Сацюк, А. В.** Автономное наведение БПЛА с использованием компьютерного зрения : проблема точного управления рулями / А. В. Сацюк, Д. В. Швалов // Автоматика на транспорте. – 2024. – Т. 10, № 4. – С. 372–381. – DOI 10.20295/2412-9186-2024-10-04-372-381.
12 **Lalak, M.** Automated Detection of Atypical Aviation Obstacles from UAV Images Using a YOLO Algorithm / M. Lalak, D. Wierzbicki // Sensors (Basel). – 2022. – Vol. 22, No. 17. – P. 6611. – DOI 10.3390/s22176611.

7 *YOLOv8*. Ultralytics. Official *YOLOv8* repository / G. Jocher, A. Stoken, A. Chaly [et al.]. – 2023. – URL: https://github.com/ultralytics/ultralytics (date of access: 02/07/2025).
8 **Zheng, Z. ???** Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation / Z. Zheng, P. Wang, D. Ren [et al.] // Transactions on Cybernetics. – DOI 10.1109/TCYB.2021.3095305.
9 An Efficient UAV-Based Aerial Image Object Detection Method via Improved YOLOv4 / B. Liu, X. Li, G. Wang [et al.] // Remote Sensing. – 2023. – Vol. 12, No. 21. – P. 3599. – DOI 10.1109/SIU53274.2021.9478027.
10 **Satsuk, A. V.** Features of the development of unmanned aerial vehicles for the railway transport industry / A. V. Satsuk, A. A. Vorobyov // Collection of scientific papers of the Donetsk Institute of Railway Transport. – 2023. – No. 68. – P. 13–21. – ISSN 1993-5579.
11 **Satsuk, A. V.** Autonomous guidance of UAVs using computer vision : the problem of precise control of rudders / A. V. Satsuk, D. V. Shvalov // Automation in transport. – 2024. – Vol. 10, No. 4. – P. 372–381. – DOI 10.20295/2412-9186-2024-10-04-372-381.
12 **Lalak, M.** Automated Detection of Atypical Aviation Obstacles from UAV Images Using a YOLO Algorithm / M. Lalak, D. Wierzbicki // Sensors (Basel). – 2022. – Vol. 22, No. 17. – P. 6611. – DOI 10.3390/s22176611.

*A. V. Satsuk*

## OPTIMIZATION OF *YOLOv8* ARCHITECTURE FOR UAV OBJECT CAPTURE TASKS: ANALYSIS OF THE TRADE-OFF BETWEEN ACCURACY, SPEED AND COMPUTATIONAL RESOURCES

**Abstract.** This paper presents a comprehensive approach to optimizing the *YOLOv8n* model for object detection tasks using unmanned aerial vehicles (UAVs) under constrained computational resources. The focus is on quantization (INT8/INT4) and pruning (50%/75%) techniques aimed at reducing the model's computational complexity while maintaining acceptable accuracy. As a result of optimization, the *YOLOv8n-Optimized-Drone* model was developed, demonstrating a 4-fold increase in processing speed on the Raspberry Pi 5 platform compared to the basic version. The model size was reduced by 3.8 times, which is critical for embedded UAV systems.

A specialized dataset with bounding box markup was created for training and validating the model, taking into account the UAV shooting conditions. Field tests confirmed the effectiveness of the proposed method, which provides a balance between performance, power consumption, and accuracy. Additionally, the influence of different quantization and pruning levels on final metrics was investigated, enabling the determination of the optimal configuration for deployment on low-power devices. The obtained results open prospects for further adaptation of the model to dynamic flight conditions and integration with multi-sensor UAV systems.

**Keywords:** unmanned aerial vehicle, quantization, pruning, neural network model optimization, neural network, *YOLO*, *YOLO metrics*.

**For citation:** Satsuk, A. V. Optimization of *YOLOv8* architecture for UAV object capture tasks: analysis of the trade-off between accuracy, speed and computational resources / A. V. Satsuk // Vestnik Rostovskogo Gosudarstvennogo Universiteta Putej Soobshcheniya. – 2025. – No. 2. – P. 35–42. – DOI 10.46973/0201–727X_2025_2_35.

**Сведения об авторах**

**Сацюк Александр Владимирович**
Донецкий институт железнодорожного транспорта (ДОНИЖТ),
кафедра «Автоматика, телемеханика, связь и вычислительная техника»,
кандидат технических наук, доцент,
e-mail: alexandrsatsuk@gmail.com

**Information about the authors**

**Satsuk Alexander Vladimirovich**
Donetsk Institute of Railway Transport (DonIRT),
Chair “Automation, Telemechanics, Communication and Computer Engineering”,
Candidate of Engineering Sciences,
Associate Professor,
e-mail: alexandrsatsuk@gmail.com