

УДК 656.073

## Практическое сравнение методов компьютерного зрения и глубокого обучения в задаче бинарной классификации изображений

<b>Лохвицкий Владимир Александрович</b>	— д-р техн. наук, заместитель начальника кафедры математического и программного обеспечения специальных космических комплексов, доцент. Научные интересы: системы массового обслуживания, разработка безопасного программного обеспечения. E-mail: lokhv_va@mail.ru
<b>Яковлев Евгений Леонидович</b>	— канд. техн. наук, доцент кафедры математического и программного обеспечения специальных космических комплексов. Научные интересы: системы искусственного интеллекта, системы управления БПЛА. E-mail: 79112249163@ya.ru
<b>Бушев Илья Владимирович</b>	— курсант 5-го курса направления 09.05.01 «Программное и математическое обеспечение систем управления летательными аппаратами». Научные интересы: цифровая обработка изображений, машинное обучение, геоинформационные системы. E-mail: bushev-ilya123@mail.ru

Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, ул. Ждановская, 13

**Для цитирования:** Лохвицкий В. А., Яковлев Е. Л., Бушев И. В. Практическое сравнение методов компьютерного зрения и глубокого обучения в задаче бинарной классификации изображений // Интеллектуальные технологии на транспорте. 2025. № 4 (44). С. 89–98. DOI: 10.20295/2413-2527-2025-444-89-98

**Аннотация.** Задачи бинарной классификации изображений широко применяются в инженерных и производственных системах, включая автоматизированный контроль, техническое зрение и мониторинг объектов. При усложнении условий съемки и увеличении объемов данных возникает необходимость сравнения классических алгоритмов компьютерного зрения и нейросетевых методов глубокого обучения с целью выбора оптимального. **Цель:** провести практическое сравнение эффективности классического алгоритма обработки изображений и нейросетевой модели YOLO при решении задачи бинарной классификации. **Методы:** традиционная обработка изображений на основе пороговой фильтрации, морфологических операций и анализа геометрических признаков, а также модель детекции YOLO, обученная на размеченной выборке. **Результаты:** классический алгоритм показал высокую скорость обработки и достаточную точность при стабильном освещении, однако продемонстрировал значительное падение качества при изменении условий съемки. Модель YOLO обеспечила более высокие показатели точности, устойчивость к фотометрическим и геометрическим вариациям, а также стабильную работу при наличии дополнительных шумов. **Практическая значимость:** результаты могут быть использованы при проектировании систем компьютерного зрения, выборе оптимального алгоритма под конкретные условия эксплуатации, а также создании гибридных систем, объединяющих преимущества классических и нейросетевых методов. **Обсуждение:** исследование подтверждает, что классические методы эффективно работают в условиях ограниченных ресурсов, но чувствительны к внешним факторам. Нейросетевые подходы, напротив, обеспечивают высокую обобщающую способность и устойчивость, что делает их предпочтительными при нестабильных условиях съемки. Новизна работы заключается в сравнении методов в идентичных условиях обработки с акцентом на практические показатели, что позволяет объективно оценить область применения каждого подхода.

**Ключевые слова:** компьютерное зрение, глубокое обучение, YOLO, бинарная классификация, анализ изображений, обработка изображений, нейросетевые методы

**1.2.1** — искусственный интеллект и машинное обучение (технические науки); **1.2.2** — математическое моделирование, численные методы и комплексы программ (технические науки); **2.3.1** — системный анализ, управление и обработка информации, статистика (технические науки)

## Введение

С развитием технологий машинного обучения, а также с ростом вычислительных мощностей значительное внимание стало уделяться сравнительному анализу методов компьютерного зрения и нейросетевых алгоритмов для решения задач обработки изображений [1]. Методы компьютерного зрения (Computer Vision, CV), такие как пороговая бинаризация и морфологическая обработка, долгое время использовались для задач, связанных с распознаванием объектов и анализом визуальной информации [2].

С увеличением объемов данных и усложнением условий съемки возникла необходимость сопоставить эффективность классических и нейросетевых методов, чтобы определить, при каких условиях каждый из них показывает лучшие результаты.

Одной из ключевых задач, решаемых с помощью указанных подходов, является бинарная классификация изображений — определение наличия или отсутствия заданного признака. Традиционные методы, основанные на пороговой обработке и морфологических операциях, остаются востребованными при ограниченных вычислительных ресурсах и стабильных условиях съемки. Однако при наличии внешних факторов — неравномерное освещение, смещение камеры, шум — их точность значительно снижается, что ставит вопрос о целесообразности применения адаптивных нейросетевых решений в условиях нестабильной съемки.

В ряде современных исследований подробно рассматриваются различия между классическими и нейросетевыми подходами.

Так, в [3] проведен сравнительный анализ методов глубокого обучения и традиционных алгоритмов компьютерного зрения в задачах идентификации дефектов. Авторы показали, что сверточные нейронные сети обеспечивают более высокую точность и автоматическую адаптацию к условиям съемки, тогда как классические алгоритмы требуют ручной настройки параметров. При этом подчеркивается, что методы глубокого обучения предъявляют повышенные требования к объему данных и вычислительным ресурсам.

В [4] рассмотрены методы и алгоритмы компьютерного зрения на основе сверточных и рекуррентных нейронных сетей. Автор отмечает, что традиционные алгоритмы, включая алгоритм Виолы — Джонса и методы на основе признаков Хаара, показывают хорошие результаты при идеализированных условиях, но теряют эффективность при изменении ракурса или освещения. Напротив, нейросетевые архитектуры, такие как YOLO, R-CNN и SSD, обеспечивают обработку изображений в реальном времени и сохраняют высокую точность при сложных условиях съемки, что делает их предпочтительными в интеллектуальных системах распознавания.

В [5] приведен обзор современных методов машинного обучения, включая контролируемое и глубокое обучение. Отмечено, что нейросетевые модели обеспечивают более высокую обобщающую способность за счет автоматического выделения признаков и нелинейного представления данных. Однако автор также указывает на проблему интерпретируемости таких моделей и необходимость большого количества размеченных данных для их обучения, что ограничивает их применение в некоторых инженерных задачах.

Таким образом, несмотря на значительный прогресс в области нейронных сетей, выбор между классическими и нейросетевыми методами обработки изображений остается актуальной задачей. Особенно важным является экспериментальное сравнение указанных подходов по критериям точности, устойчивости к внешним факторам и вычислительной эффективности. В рамках настоящей работы проводится практический анализ двух подходов — классического метода обработки изображений на основе пороговой бинаризации и морфологических операций и модели YOLO. Сравнение осуществляется по показателям точности, устойчивости к изменениям освещенности и времени обработки изображений.

## Материалы и методы

Рассматривается задача бинарной классификации изображений, в рамках которой необходимо

определить принадлежность входного изображения  $I$  к одному из двух классов: наличие или отсутствие целевого объекта. Для решения задачи анализируются два подхода — классический алгоритм компьютерного зрения, основанный на пороговой бинаризации и морфологических операциях, и нейросетевая модель YOLO, обученная на размеченных данных.

**Дано:**

1. Входное изображение:

$$I \in \mathbb{R}^{H \times W \times C},$$

где  $H$  — количество пикселей по вертикали;

$W$  — количество пикселей по горизонтали;

$C$  — количество цветовых каналов.

2. Для классического подхода используется функция предобработки:

$$F_{cv}(I) = B(I, T),$$

где  $B(I, T)$  — бинаризованное изображение, полученное по порогу  $T$ .

Порог  $T$  выбирается эмпирически и определяет чувствительность алгоритма к изменениям яркости. После бинаризации применяются морфологические операции для устранения шумов и выделения ключевых областей.

3. Для нейросетевого подхода используется обученная модель

$$F_{yolo}(I; \theta),$$

где  $\theta$  — набор параметров нейронной сети, определяющий ее способность выделять и классифицировать объекты на изображении.

4. Результатом работы модели является вектор вероятностей:

$$P = \{p_0, p_1\}, \quad p_0 + p_1 = 1,$$

где  $p_1$  — вероятность наличия целевого объекта на изображении;

$p_0$  — вероятность отсутствия целевого объекта на изображении.

**Требуется определить:**

1. Класс изображения:

$$y = \arg \max_{k \in \{0,1\}} P_k,$$

где  $y = 1$  — объект присутствует;

$y = 0$  — объект отсутствует.

2. Метрики качества работы методов:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

где  $\text{TP}$ ,  $\text{TN}$ ,  $\text{FP}$ ,  $\text{FN}$  — числа истинно и ложно классифицированных изображений;

$\text{Accuracy}$  — общая точность;

$\text{Precision}$  — точность положительных срабатываний;

$\text{Recall}$  — доля реально существующих объектов, которые выявил алгоритм.

3. Оценить влияние внешних факторов на качество классификации каждого подхода.

## Методы исследования

Предлагаемый сравнительный анализ реализует два независимых алгоритма обработки изображений, направленных на решение задачи бинарной классификации:

1. Классический алгоритм компьютерного зрения, основанный на пороговой бинаризации и морфологических операциях.

2. Нейросетевой алгоритм на основе архитектуры YOLO.

Оба подхода обеспечивают определение принадлежности изображения к классу «объект присутствует/отсутствует», но используют принципиально различные способы извлечения признаков и принятия решения.

## Классический алгоритм компьютерного зрения

Алгоритм основан на последовательной обработке изображения с применением базовых методов цифровой обработки и морфологического анализа. Он включает следующие ключевые этапы:

1. **Предобработка изображений:** изображение  $I(x, y)$  приводится к оттенкам серого и нормализуется [6]:

$$I_g(x, y) = 0,299R(x, y) + 0,587G(x, y) + \\ + 0,114B(x, y),$$

где  $R, G, B$  — значения каналов в координате  $(x, y)$ .

Далее применяется фильтрация для подавления шумов. Используется, например, гауссов фильтр:

$$I_f(x, y) = G_y * I_g(x, y),$$

где  $G_y$  — двумерное гауссово ядро с дисперсией  $\sigma$ ;  $*$  — операция свертки.

**2. Пороговая бинаризация:** для выделения областей интереса применяется операция пороговой бинаризации [7]:

$$B(x, y) = \begin{cases} 1, & \text{если } I_f(x, y) > T; \\ 0, & \text{если } I_f(x, y) \leq T, \end{cases}$$

где порог  $T$  выбирается эмпирически и определяет чувствительность к изменению яркости.

**3. Морфологическая обработка:** применяется для удаления и объединения смежных участков [8]:

$$B'(x, y) = D(E(B(x, y))),$$

где  $E(\cdot)$  — операция эрозии;

$B(\cdot)$  — операция дилатации.

**4. Извлечение признаков и классификация:** из бинарной маски вычисляются геометрические характеристики: площадь  $S$ , периметр  $P$ , соотношение сторон, компактность и т. д. На основе этих признаков определяется наличие или отсутствие объекта:

$$y_{cv} = \begin{cases} 1, & \text{если } S > S_{min} \wedge P > P_{min}; \\ 0, & \text{в противном случае.} \end{cases}$$

### Нейросетевой алгоритм

Нейросетевая модель YOLO представляет собой архитектуру сверточной нейронной сети, выполняющую детекцию и классификацию объектов в едином проходе изображения. Алгоритм включает следующие этапы:

**1. Подготовка данных:** исходные изображения приводятся к фиксированному размеру  $416 \times 416$  пикселей и нормализуются в диапазон  $[0, 1]$ :

$$I_n(x, y, c) = \frac{255}{I(x, y, c)}.$$

Для обучения формируется размеченная выборка  $\{(I_i, y_i)\}_{i=1}^N$ ,

где  $y_i \in \{0, 1\}$  — метка класса для изображения  $I_i$ .

**2. Архитектура и обучение:** YOLO делит изображение на сетку размером  $S \times S$ , для каждой ячейки предсказывает наличие объекта и параметры ограничивающего прямоугольника. Функция потерь описывается выражением:

$$L = \lambda_{coord} \sum_{i=0}^{S^2} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \\ + \lambda_{obj} \sum_{i=0}^{S^2} (C_i - \hat{C}_i)^2 + \\ + \lambda_{cls} \sum_{i=0}^{S^2} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2,$$

где  $(x_i, y_i)$  — координаты центра объекта;

$C_i$  — вероятность наличия объекта в ячейке;

$p_i(c)$  — вероятность принадлежности к классу  $C$ ;

$\lambda_{coord}, \lambda_{obj}, \lambda_{cls}$  — весовые коэффициенты ошибок.

**3. Распознавание:** после обучения модель принимает изображение  $I$  и выдает вектор вероятностей:

$$P = F_{yolo}(I; \theta) = \{p_0, p_1\},$$

где  $\theta$  — обученные веса сети.

Класс определяется следующим образом:

$$y_{yolo} = \arg \max_{k \in \{0, 1\}} p_k.$$

Для повышения точности применяется процедура подавления немаксимальных значений (Non-Maximum Suppression, NMS), устраняющая дублирующиеся предсказания.

### Сравнительный анализ

Для объективной оценки эффективности методов выполняется сравнение по трем ключевым критериям:

- точность классификации — по метрике Accuracy;
- устойчивость к внешним воздействиям — при варьировании освещенности, угла наклона и уровня шума;
- время обработки одного изображения, определяемое экспериментально.

Сравнение производится на идентичном наборе изображений, что обеспечивает равные условия для обоих алгоритмов.

### Реализация и экспериментальная часть

Для реализации и последующего сравнения был разработан программный комплекс с графическим интерфейсом, построенный по модульной архитектуре [9]. Каждый модуль решает специализированную задачу, обеспечивая последовательное выполнение операций — от загрузки изображений до анализа и визуализации результатов (табл. 1). Такое построение отражает принципы модульности и воспроизводимости экспериментов, что позволяет выполнять корректное сравнение алгоритмов в идентичных условиях [10].

Система реализована в виде набора модулей, каждый из которых выполняет строго определенные функции. Такое разделение обеспечивает независимость компонентов, возможность последующего расширения функционала и удобство тестирования.

Для проверки теоретических положений, описанных ранее, была проведена практическая оценка эффективности двух подходов — классического и нейросетевого [15].

Тестирование проводилось на едином наборе изображений при идентичных условиях обработки. Для обеих систем использовались одинаковые параметры предобработки и фиксированный размер входного кадра. Метрики качества вычислялись на тестовой части выборки по стандартным формулам, принятым в задачах бинарной классификации [16]. Основное внимание уделялось трем группам характеристик:

1. Качественные показатели: Accuracy, Precision, Recall.

2. Производительность и вычислительная эффективность: время обработки, FPS.

3. Устойчивость к внешним факторам: изменения освещенности, шум, геометрические искажения.

Для оценки точности классификации были вычислены основные показатели качества — Accuracy, Precision, Recall и F1 [7]. Результаты сравнения эффективности двух подходов представлены в табл. 2.

Классический алгоритм демонстрирует стабильность при контролируемом освещении, однако при снижении контрастности теряет часть истинных срабатываний. Нейросетевая модель, напротив, сохраняет высокие значения метрик за

Таблица 1

#### Основные этапы работы программного комплекса

Этап	Описание	Используемые инструменты
Загрузка изображений	Импорт входных изображений и приведение их к унифицированному формату	OpenCV [11], NumPy [12]
Предобработка	Конвертация в оттенки серого, фильтрация шумов, нормализация размеров	OpenCV
Анализ методом CV	Пороговая бинаризация и морфологические операции для выделения областей интереса	OpenCV
Анализ методом YOLO	Обработка изображений обученной нейросетевой моделью YOLO, получение вероятностных карт	PyTorch [13], Ultralytics
Сравнение результатов	Расчет метрик точности, полноты и времени обработки	NumPy, time
Визуализация	Отображение исходных и результирующих изображений, построение графиков	PyQt5, Matplotlib [14]

Таблица 2

#### Результаты оценки качества классификации

Метрика	CV	YOLO
Accuracy	0,87	0,98
Precision	0,82	0,97
Recall	0,79	0,98
F1	0,80	0,97

счет обучения на аугментированных данных и способности обобщать признаки.

Скорость обработки изображений является ключевым параметром при внедрении систем реального времени. Для количественной оценки производительности обоих методов были проведены измерения времени обработки кадров и загрузки вычислительных ресурсов (табл. 3). Эксперименты выполнялись на едином аппаратном стенде: CPU Intel Core i7-12700H, 16 GB RAM, GPU RTX 3070 Ti.

Как следует из табл. 3, классический метод выигрывает по скорости на CPU-платформах, однако YOLO демонстрирует стабильную производительность при использовании GPU, обеспечивая обработку кадров в режиме, близком к реальному времени.

Для анализа устойчивости алгоритмов к внешним воздействиям были сформированы дополнительные тест-наборы с варьированием условий: изменение освещенности, добавление гауссова шума и поворот изображения. Результаты измерений точности при различных сценариях приведены в табл. 4.

Как видно из табл. 4, CV-метод значительно чувствительнее к изменению освещенности и шуму, поскольку основан на фиксированных порогах яркости. YOLO демонстрирует высокую устойчивость за счет обученных сверточных фильтров и встроенных аугментаций, обеспечивающих инвариантность к фотометрическим и геометрическим вариациям.

Проведенные испытания подтвердили, что нейросетевой подход обеспечивает более высокую точность, полноту и устойчивость при разнообразных условиях съемки. Классические методы, в свою очередь, сохраняют актуальность в системах с ограниченными вычислительными ресурсами, где приоритетом являются высокая скорость обработки и простота интерпретации решений.

Таким образом, выбор подхода определяется балансом между требуемой точностью, ресурсными возможностями и характером решаемой задачи. В случаях, когда важна объяснимость алгоритма и минимальная нагрузка на процессор, предпочтителен классический метод. При необходимости высокой точности и устойчивости к внешним факторам оптимальным выбором является использование нейросетевой модели YOLO.

## Заключение

Проведенное исследование подтвердило, что выбор метода анализа изображений определяется сочетанием факторов — требуемой точности, устойчивости к внешним воздействиям и доступных вычислительных ресурсов. Реализованный программный комплекс позволил выполнить корректное сравнение двух подходов — классического алгоритма компьютерного зрения и нейросетевой модели YOLO — в идентичных условиях обработки.

Результаты экспериментов показали, что классический метод, основанный на пороговой бина-

Таблица 3

### Сравнение производительности алгоритмов

Показатель	CV	YOLO	Примечание
Среднее время на кадр, мс	4,6	11,3	YOLO выполняет больше вычислений
FPS, кадров/с	217	88	CV в 2,4 раза быстрее
Использование CPU/GPU	CPU 100 %	GPU 35 % / CPU 20 %	YOLO эффективно распараллеливается
Память, МБ	85	312	Веса модели и активации

Таблица 4

### Сравнение производительности алгоритмов

Условие	CV — Accuracy	YOLO — Accuracy	ΔAcc (CV)	ΔAcc (YOLO)
Базовый набор	0,87	0,98	—	—
Освещенность ±20 %	0,74	0,96	-0,13	-0,02
Шум ( $\sigma = 10$ )	0,71	0,95	-0,16	-0,03
Поворот ±10°	0,77	0,97	-0,10	-0,01

ризации и морфологических операциях, обладает высокой скоростью обработки и простотой реализации, что делает его эффективным инструментом при стабильных условиях съемки и ограниченных ресурсах вычислительной системы. Однако его точность снижается при изменении освещенности, наличии шумов и геометрических искажениях изображения.

Нейросетевая модель YOLO, напротив, демонстрирует более высокие показатели точности, полноты и устойчивости к вариациям входных данных за счет обучаемых признаков и способности к адаптации, но предъявляет повышенные требования к объему обучающих данных и использованию графических ускорителей. Полученные результаты не свидетельствуют об абсолютном превосходстве одного подхода над другим, а от-

ражают различие в приоритетах: классические алгоритмы обеспечивают лучшую скорость и простоту настройки, тогда как нейросетевые методы — устойчивость и точность. На практике целесообразно выбирать метод с учетом особенностей задачи, вычислительных ресурсов и требований к интерпретируемости.

В перспективе результаты исследования могут быть использованы для разработки гибридных систем, объединяющих преимущества обоих подходов [17]. Такие решения позволяют реализовать адаптивную обработку изображений, при которой предварительная фильтрация выполняется классическими методами, а уточняющая классификация — с использованием нейронных сетей, обеспечивая баланс между точностью и вычислительной эффективностью.

## СПИСОК ИСТОЧНИКОВ

1. Сеничев А. В., Новикова А. И., Васильев П. В. Сравнение глубокого обучения с традиционными методами компьютерного зрения в задачах идентификации дефектов // Молодой исследователь Дона. 2020. № 4 (25). С. 64–67.
2. Полковникова Н. А. Исследование методов и алгоритмов компьютерного зрения на основе сверточных и рекуррентных нейронных сетей // Эксплуатация морского транспорта. 2020. № 3 (96). С. 154–168. DOI: 10.34046/aumsuomt96/21.
3. Саксонов П. В., Баuman А. А. Обзор методов машинного обучения // Современные тенденции и инновации в науке и производстве: материалы XII Международной научно-практической конференции (Междуреченск, Россия, 26 апреля 2023 г.). Междуреченск: Кузбасский гос. технический ун-т им. Т. Ф. Горбачева, 2023. С. 444-1–444-6.
4. Гонсалес Р. С., Вудс Р. Е. Цифровая обработка изображений = Digital Image Processing. Third Edition / пер. с англ. Л. И. Рубанова и П. А. Чочиа. 3-е изд., испр. и доп. М.: Техносфера, 2012. 1104 с.
5. Сеничев А. В., Новиков С. С. Методы компьютерного зрения и их применение в анализе изображений. М.: ИПРАН РАН, 2020. 142 с.
6. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol (CA): O'Reilly Media, 2008. 575 p.
7. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement // ArXiv. 2018. Vol. 1804.02767. 6 p. DOI: 10.48550/arXiv.1804.02767.
8. Caballar R. D., Stryker C. What is Computer Vision? URL: <http://www.ibm.com/think/topics/computer-vision> (дата обращения: 07.11.2025).
9. Patel M. The Complete Guide to Image Preprocessing Techniques in Python. URL: <http://readmedium.com/the-complete-guide-to-image-preprocessing-techniques-in-python-dca30804550c> (дата обращения: 07.11.2025).
10. Thresholding (Image Processing) // Wikipedia. Last update 12 November 2025. URL: [http://en.wikipedia.org/wiki/Thresholding\\_\(image\\_processing\)](http://en.wikipedia.org/wiki/Thresholding_(image_processing)) (дата обращения: 16.11.2025).
11. Mathematical Morphology // Wikipedia. Last update 08 November 2025. URL: [http://en.wikipedia.org/wiki/Mathematical\\_morphology](http://en.wikipedia.org/wiki/Mathematical_morphology) (дата обращения: 16.11.2025).
12. Bradski G. The OpenCV Library // Dr. Dobb's Journal Software Tools. 2000. Vol. 25, Iss. 11. Pp. 120–125.

13. PyTorch: An Imperative Style, High-Performance Deep Learning Library / A. Paszke, S. Gross, F. Massa [et al.] // Advances in Neural Information Processing Systems 32: Proceeding of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), (Vancouver, Canada, 08–14 December 2019). NeurIPS Foundation, 2019. Pp. 8024–8035.
14. Hunter J. D. Matplotlib: A 2D Graphics Environment // Computing in Science and Engineering. 2007. Vol. 9, Iss. 3. Pp. 90–95. DOI: 10.1109/MCSE.2007.55.
15. Turay T., Vladimirova T. Toward Performing Image Classification and Object Detection with Convolutional Neural Networks in Autonomous Driving Systems: A Survey // IEEE Access. 2022. Vol. 10. Pp. 14076–14119. DOI: 10.1109/ACCESS.2022.3147495.
16. Powers D. M. W. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation // Journal of Machine Learning Technologies. 2011. Vol. 2, Iss.1. Pp. 37–63.
17. Клэтте Р. Компьютерное зрение. Теория и алгоритмы = Concise Computer Vision. An Introduction into Theory and Algorithms / пер. с англ. А. А. Слинкина. М.: ДМК Пресс, 2019. 506 с.

Дата поступления: 20.11.2025

Решение о публикации: 25.11.2025

## Practical Comparison of Computer Vision and Deep Learning Methods for the Binary Image Classification Task

- Vladimir A. Lokhvitsky** — Dr. Sci. in Engineering, Deputy Head of the Department of Mathematics and Software for Special Space Complexes, Associate Professor. Research interests: queuing systems, secure software development. E-mail: lokhv\_va@mail.ru
- Evgeny L. Yakovlev** — PhD in Engineering, Associate Professor, Department of Mathematics and Software for Special Space Complexes. Research interests: artificial intelligence systems, UAV control systems. E-mail: 79112249163@ya.ru
- Ilya V. Bushev** — 5th year Student Majoring in 09.05.01 Software and Mathematical Support for Aircraft Control Systems. Research interests: digital image processing, machine learning, geographic information systems. E-mail: bushev-ilya123@mail.ru

Mozhaisky Military Aerospace Academy, 13, Zhdanovskaya str., Saint Petersburg, 197198, Russia

**For citation:** Lokhvitsky V. A., Yakovlev E. L., Bushev I. V. Practical Comparison of Computer Vision and Deep Learning Methods for the Binary Image Classification Task. *Intellectual Technologies on Transport*, 2025, No. 4 (44), Pp. 89–98. DOI: 10.20295/2413-2527-2025-444-89-98. (In Russian)

**Abstract.** *Binary image classification tasks are widely employed in engineering and manufacturing systems, including automated control, machine vision, and object monitoring. As the complexity of imaging conditions increases and data volumes expand, it becomes essential to evaluate classical computer vision algorithms alongside deep learning neural network methods to identify the most effective method. Purpose: to perform a practical comparison of the effectiveness of a traditional image processing algorithm and the YOLO neural network model for addressing the binary classification challenge. Methods: traditional image processing techniques, including threshold filtering, morphological operations, and geometric feature analysis, as well as the YOLO detection model trained on a labeled dataset. Results: the classical algorithm has demonstrated high processing speed and adequate accuracy under stable lighting conditions; however, it has exhibited a pronounced decline in performance when imaging conditions changed. In contrast, the YOLO model has demonstrated enhanced accuracy, resilience to both photometric and geometric variations, and consistent performance even in the presence of extraneous noise. Practical significance: the results can guide the development of computer vision systems, aid in the selection of the optimal algorithm for specific operational scenarios, and*

facilitate the creation of hybrid systems that combine the strengths of both traditional and neural network methods. **Discussion:** this study confirms that while traditional methods are effective in resource-constrained environments, they are vulnerable to external influences. Conversely, neural network approaches offer superior generalization and stability, making them more advantageous in fluctuating imaging conditions. The novelty of the research lies in the comparative analysis of these methods under identical processing parameters, with a particular focus on practical indicators. This approach facilitates an objective assessment of the applicability domain inherent to each method.

**Keywords:** computer vision, deep learning, YOLO, binary classification, image analysis, image processing, neural network methods

## REFERENCES

1. Senichev A. V., Novikova A. I., Vasilyev P. V. Sravnenie glubokogo obucheniya s traditsionnymi metodami kompyuternogo zreniya v zadachakh identifikatsii defektov [Comparison of Deep Learning with Traditional Methods of Computer Vision in the Problems of Defects Identification], *Molodoy issledovatel Dona* [Young Researcher of Don], 2020, No. 4 (25), Pp. 64–67. (In Russian)
2. Polkovnikova N. A. Issledovanie metodov i algoritmov kompyuternogo zreniya na osnove svertochnykh i rekurrentnykh neyronnykh setey [Research of Methods and Algorithms of Computer Vision Based on Convolutional and Recurrent Neural Networks], *Ekspluatatsiya morskogo transporta* [Marine Transport Operation], 2020, No. 3 (96), Pp. 154–168. DOI: 10.34046/aumsuomt96/21. (In Russian)
3. Saksonov P. V., Bauman A. A. Obzor metodov mashinnogo obucheniya [Overview of Machine Learning Methods], *Sovremennye tendentsii i innovatsii v nauke i proizvodstve: materialy XII Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Modern Trends and Innovations in Science and Production: Proceedings of the XII International Scientific and Practical Conference (Mezhdurechensk, Russia, April 26, 2023)], Mezhdurechensk, T. F. Gorbachev Kuzbass State Technical University, 2023. Pp. 444-1–444-6. (In Russian)
4. Gonzalez R. C., Woods R. E. Tsifrovaya obrabotka izobrazheniy [Digital Image Processing. Third Edition]. Moscow, Tekhnosphera Publishing House, 2012, 1104 p. (In Russian)
5. Senichev A. V., Novikov S. S. Metody kompyuternogo zreniya i ikh primenie v analize izobrazheniy [Computer vision methods and their application in image analysis]. Moscow, Institute for the Study of Science of the RAS, 2020, 142 p. (In Russian)
6. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol (CA), O'Reilly Media, 2008, 575 p.
7. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement, *ArXiv*, 2018, Vol. 1804.02767, 6 p. DOI: 10.48550/arXiv.1804.02767.
8. Caballar R. D., Stryker C. What is Computer Vision? Available at: <http://www.ibm.com/think/topics/computer-vision> (accessed: November 07, 2025).
9. Patel M. The Complete Guide to Image Preprocessing Techniques in Python. Available at: <http://readmedium.com/the-complete-guide-to-image-preprocessing-techniques-in-python-dca30804550c> (accessed: November 07, 2025).
10. Thresholding (Image Processing), *Wikipedia*. Last update November 12, 2025. Available at: [http://en.wikipedia.org/wiki/Thresholding\\_\(image\\_processing\)](http://en.wikipedia.org/wiki/Thresholding_(image_processing)) (accessed: November 16, 2025).
11. Mathematical Morphology, *Wikipedia*. Last update November 08, 2025. Available at: [http://en.wikipedia.org/wiki/Mathematical\\_morphology](http://en.wikipedia.org/wiki/Mathematical_morphology) (accessed: November 16, 2025).
12. Bradski G. The OpenCV Library, *Dr. Dobb's Journal Software Tools*, 2000, Vol. 25, Iss. 11, Pp. 120–125.
13. Paszke A., Gross S., Massa F., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library, *Advances in Neural Information Processing Systems 32: Proceeding of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, December 08–14, 2019*, NeurIPS Foundation, 2019. Pp. 8024–8035.

14. Hunter J. D. Matplotlib: A 2D Graphics Environment, *Computing in Science and Engineering*, 2007, Vol. 9, Iss. 3. Pp. 90–95. DOI: 10.1109/MCSE.2007.55.
15. Turay T., Vladimirova T. Toward Performing Image Classification and Object Detection with Convolutional Neural Networks in Autonomous Driving Systems: A Survey, *IEEE Access*, 2022, Vol. 10, Pp. 14076–14119. DOI: 10.1109/ACCESS.2022.3147495.
16. Powers D. M. W. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation, *Journal of Machine Learning Technologies*, 2011, Vol. 2, Iss.1, Pp. 37–63.
17. Klette R. Kompyuternoe zrenie. Teoriya i algoritmy [Concise Computer Vision. An Introduction into Theory and Algorithms]. Moscow, DMK Press Publishing House, 2019, 506 p. (In Russian)

Received: 20.11.2025

Accepted: 25.11.2025