

УДК 004.896

DOI: [10.26102/2310-6018/2026.54.3.020](https://doi.org/10.26102/2310-6018/2026.54.3.020)

## Исследование нейронных сетей как способа сжатия и архивации изображений

Я.Д. Лоскутов, Д.А. Грецкий, Ц.Б. Пронин, А.А. Подберезкин✉, А.В. Остроух

*Московский автомобильно-дорожный государственный технический университет,  
Москва, Российская Федерация*

**Резюме.** В статье исследован метод хранения изображений путем обучения нейронной сети на одном изображении и сохранения ее весов как компактного представления. Данный подход позволяет значительно уменьшить объем хранимых данных при сохранении приемлемого визуального качества. Проанализированы параметры модели и настройки обучения с целью оптимизации качества восстановления. Основная идея подхода заключается в том, что обученная модель сохраняет аппроксимированное отображение изображения в свои веса, которые выступают в качестве компактного представления исходного изображения. При необходимости реконструкции веса загружаются обратно в сеть для восстановления визуального содержания. Экспериментальные результаты показывают, что оптимизация архитектуры сети и цветового пространства (YCbCr) позволяет достичь высоких коэффициентов сжатия до 29,4 при сохранении визуального качества, близкого к оригиналу ( $MSE \approx 10^{-5}$ ). Однако авторы отмечают существенный недостаток метода: длительное время обучения и значительные вычислительные затраты, что делает его менее эффективным по сравнению с традиционными алгоритмами сжатия для практического применения в реальном времени. Тем не менее, подход демонстрирует потенциал для задач, где критически важно сохранение тонких деталей изображений, например, архивации данных или сжатия видеопотоков.

**Ключевые слова:** сжатие изображений, нейронная сеть, архивация изображений, обучение на одном изображении, восстановление изображения, многослойный перцептрон, машинное обучение, позиционное кодирование, координатное кодирование, искусственный интеллект.

**Для цитирования:** Лоскутов Я.Д., Грецкий Д.А., Пронин Ц.Б., Подберезкин А.А., Остроух А.В. Исследование нейронных сетей как способа сжатия и архивации изображений. *Моделирование, оптимизация и информационные технологии*. 2026;14(3). URL: <https://moitvivr.ru/ru/journal/article?id=2247> DOI: 10.26102/2310-6018/2026.54.3.020

## Research into neural networks as a method for image compression and archiving

Ya.D. Loskutov, D.A. Gretsii, C.B. Pronin, A.A. Podberezkin✉, A.V. Ostroukh

*Moscow Automobile and Road Construction State Technical University, Moscow,  
the Russian Federation*

**Abstract.** This article explores a method for storing images by training a neural network on a single image and storing its weights as a compact representation. This approach significantly reduces the amount of data stored while maintaining acceptable visual quality. Model parameters and training settings are analyzed to optimize recovery quality. The basic idea of the approach is that a trained model stores its weights, which act as a compact representation of the original image. When reconstruction is required, the weights are reloaded into the network to restore the visual content. Experimental results show that optimizing the network architecture and color space (YCbCr) enables high compression ratios – up to 29.4 while maintaining visual quality close to the original ( $MSE \approx 10^{-5}$ ). However, the authors note a significant drawback of the method: long training time and significant computational costs, making it less effective than traditional compression algorithms for practical real-time applications.

Nevertheless, the approach demonstrates potential for tasks where preserving fine image details is critical, such as data archiving or video stream compression.

**Keywords:** image compression, neural network, image archiving, single-image training, image restoration, multilayer perceptron, machine learning, positional coding, coordinate coding, artificial intelligence.

**For citation:** Loskutov Ya.D., Gretsii D.A., Pronin C.B., Podberezkin A.A., Ostroukh A.V. Research into neural networks as a method for image compression and archiving. *Modeling, Optimization and Information Technology*. 2025;13(4). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2247> DOI: 10.26102/2310-6018/2026.54.3.020

## Введение

Современные цифровые изображения имеют большой объем данных, что создает проблемы при передаче по сети и хранении. Существующие алгоритмы сжатия позволяют значительно уменьшить размер файла практически без заметной потери качества, однако дальнейшее сжатие все равно приводит к искажениям. Указанные ограничения определяют целесообразность применения методов искусственного интеллекта и машинного обучения, способных обеспечивать более эффективное представление и восстановление изображений. Актуальность использования таких подходов в прикладных инженерных системах подтверждается результатами недавних исследований [1, 2]. Ранее предпринимались попытки в аналогичном направлении, в частности, в работе [3] разработан сквозной фреймворк сжатия изображений на базе сверточных нейронных сетей (CNN), который включает ComCNN для генерации компактного представления входного изображения и RecCNN для его высококачественной реконструкции. В этом исследовании предлагается альтернативный подход: использовать нейронные сети для «запоминания» и последующего восстановления изображения. Идея заключается в том, что можно обучить нейросеть на одном изображении (подавая на вход только этот снимок) и затем сохранить веса сети как сжатое представление изображения. При этом фактические цвета и яркость каждого пикселя не сохраняются напрямую, а восстанавливаются самой сетью при инференсе.

На Рисунке 1 показано исходное тестовое изображение большой емкости (15606,9 Кб), сжатое для помещения в текст статьи классическим методом. Его свойства и качество заданы заранее, и именно оно используется в качестве контрольного примера при оценке предложенного метода. Для сравнения: традиционные алгоритмы сжатия позволяют уменьшить размер этого же изображения до ~1,470 КБ без заметной потери визуального качества (40 % от исходного качества), при сжатии ниже этой отметки проявляются артефакты сжатия в виде потери цветопередачи и деталей. Предлагаемый метод сжатия весов с помощью нейронной сети не воспроизводит исходное изображение пиксель в пиксель, но дает возможность высокоэффективно сохранять информацию о нем.

В качестве потенциальных приложений рассматриваются, например, предпросмотр изображений и передача через каналы с низкой пропускной способностью. Исходное изображение используется для оценки качества восстановления. Поскольку при значении функции потерь ниже  $10^{-3}$  разницу в качестве визуально заметить довольно сложно, для тестов рассчитана средняя абсолютная ошибка (Mean Absolute Error, MAE), которая колеблется в диапазоне от 0,01 до 0,009. В рамках данного исследования предполагается достижение целевого значения функции потерь порядка  $10^{-5}$ , при котором получаемое качество восстановления оказывается сопоставимым с оригинальным изображением. Современные подходы к сжатию включают обучение с квантованием весов и активаций [4], что также актуально для

нашей задачи по сохранению компактности моделей, но может привести к более существенным потерям качества.

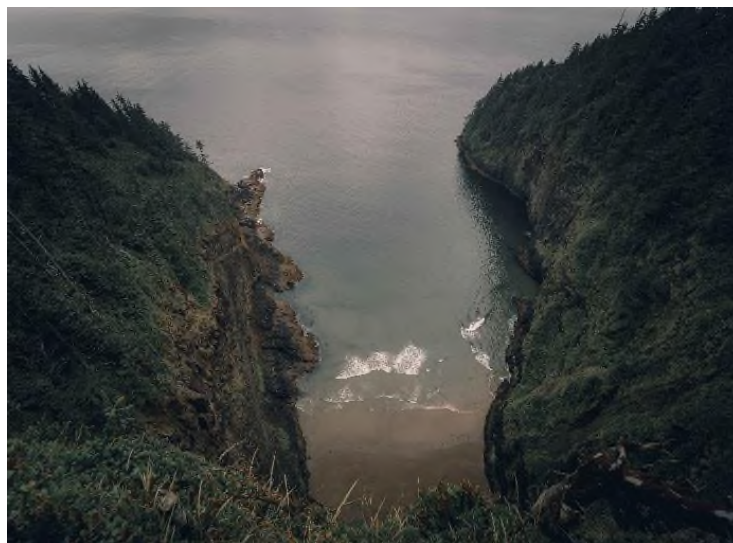


Рисунок 1 – Пример исходного изображения, сжатого классическим способом  
Figure 1 – An example of the original image compressed using a classical method

В процессе обучения учитываются следующие ключевые параметры нейросети:

- слои кодирования (число уровней кодирования входных данных, определяющее степень их предварительного преобразования);
- нейроны (число скрытых нейронов);
- слои нейронов (число нейронных слоёв в архитектуре сети);
- эпохи (число эпох обучения);
- размер пакета (размер пакета данных при обучении).

Также учитывается выбор цветового пространства входного изображения.

Задачей исследования является анализ влияния этих параметров на качество восстановленного изображения и размер файла с весами нейронной сети.

### Материалы и методы

Использование многослойного персептрона для сжатия изображений также рассматривалось в научных трудах [5], однако здесь мы используем специфическую технику позиционного кодирования. В предложенном методе использована архитектура на основе многослойного персептрона (MLP) с позиционным кодированием входных координат пикселей. В отличие от классических автоэнкодеров, сеть обучалась отображать нормализованные координаты пикселей  $(x, y)$  непосредственно в значения цветовых каналов.

Исходное изображение имеет размер  $W \times H$ . Для пикселя с индексами  $i, j$  вычисляются нормализованные координаты:  $x = \frac{i}{W-1}$ ,  $y = \frac{j}{H-1}$ . Целевыми значениями служит вектор цветовых компонент пикселя (нормализованные в диапазоне  $[0, 1]$ ).

Каждый нормализованный скаляр  $v$  (то есть  $x$  или  $y$ ) преобразуется в вектор вида:

$$\left[ v, \sin 2^{0\pi v}, \cos 2^{0\pi v}, \sin 2^{1\pi v}, \cos 2^{1\pi v}, \dots, \sin 2^{(L-1)\pi v}, \cos 2^{(L-1)\pi v} \right],$$

где  $L$  – число уровней кодирования. Кодирование  $x$  и  $y$  объединяется в единый входной вектор для сети. Такое частотное расширение входа позволяет MLP аппроксимировать высокочастотные детали изображения без экспоненциального роста числа параметров.

Сеть представляет собой многослойный перцептрон  $f\theta$ , который принимает на вход позиционно закодированный вектор координат  $(x, y)$  и выдает трехкомпонентный цветовой вектор для соответствующего пикселя. Внутри сети выполняются последовательные линейные преобразования с нелинейными активациями типа ReLU; на выходе применяется поэлементное ограничение, обеспечивающее значения в диапазоне  $[0, 1]$ . Количество слоев и число нейронов в слоях определяют емкость модели и влияют на число обучаемых параметров  $\theta$ , а следовательно – на итоговый размер файла весов.

В качестве функции потерь используется среднеквадратичная ошибка (MSE) между предсказанными и истинными цветовыми векторами по набору пикселей. Обучение производится стохастической пакетной оптимизацией с адаптивным методом (Adam) и схемой затухания шага обучения. В то время как мы используем адаптивный метод Adam, генетические алгоритмы также успешно применяются для оптимизации параметров в задачах сжатия [6, 7]. Размер пакета, число эпох и параметры оптимизатора определяют скорость сходимости и стабильность обучения. В прикладных задачах машинного обучения особое значение имеют процедуры валидации и контроля переобучения, обеспечивающие проверку устойчивости получаемых решений и корректную оценку качества моделей [1]. Аналогичные закономерности влияния настройки операторов и параметров оптимизации на сходимость и устойчивость решений отмечаются и для эволюционных методов оптимизации, где выбор операторов и их параметров существенно влияет на скорость и характер сходимости [6, 7].

После обучения параметры  $\theta$  сохраняются в формате float32, затем приводятся к float16 и архивируются. Отношение размера исходного массива параметров к размеру сжатого файла даёт коэффициент сжатия. Комбинация приведения точности и архивирования позволяет существенно уменьшить объём итогового файла весов.

Для восстановления изображения размера  $W \times H$  сеть прогоняется по всем координатам сетки: для каждой пары  $(i, j)$  вычисляются  $x = \frac{i}{W-1}$ ,  $y = \frac{j}{H-1}$ , строится позиционно закодированный вход и вычисляется выход сети – три компоненты цвета, которые записываются как пиксель восстановленного изображения. Таким образом файл весов выступает в роли компактного «кода» изображения, а прогон MLP – в роли декодера.

В ходе обучения нейросети минимизируется функция потерь (loss), измеряющая разницу между входным и выходным (восстановленным) изображениями. После достижения заданного качества (низкого значения loss) файл с весами обученной модели сохраняется и дополнительно сжимается стандартным алгоритмом архивирования. Восстановление изображения осуществляется так: архивированный файл весов предварительно распаковывается, после чего веса загружаются в сеть точно с такой же архитектурой, которая применялась при обучении, и сеть выдаёт аппроксимированное восстановление исходного изображения<sup>1</sup>.

В процессе эксперимента варьировались настройки сети и обучающего процесса. Для предобработки входных данных использовались различные уровни кодирования, при увеличении которых усиливался контраст между соседними пикселями, что облегчало нейросети запоминание структуры изображения. Число нейронов в скрытых слоях и количество нейронных слоев определяли емкость модели и напрямую влияли на размер файла весов. Параметры обучения, такие как число эпох и размер пакета, определяли скорость сходимости и стабильность оптимизации. Дополнительно исследовалось влияние выбора цветового пространства входного изображения (RGB,

<sup>1</sup> telkuzel. neuro-image-compression. GitHub. URL: <https://github.com/telkuzel/neuro-image-compression> (дата обращения: 22.02.2026).

LAB и YCbCr) на качество восстановления и размер итогового файла весов. Точность передачи цветовой информации является критически важным фактором для прикладных задач цифровизации инфраструктур [1].

### Результаты

В первоначальной программной реализации был использован фреймворк TensorFlow 2. В Таблице 1 приведены примеры экспериментальных результатов при различном сочетании параметров сети (уровней кодирования, числа нейронов, слоев кодирования, числа эпох и размера пакета) с указанием полученного значения loss и размера файла с весами (до архивирования). Анализ показал, что на качество восстановления (loss) наибольшее влияние оказывает число скрытых нейронов и размер пакета: слишком большое число нейронов существенно замедляет обучение и приводит к очень большому файлу весов, что противоречит цели сжатия, тогда как слишком маленький пакет делает обучение медленным. Количество эпох после определенного порога дает лишь незначительный прирост в качестве, но при малом числе эпох loss может значительно снижаться. Параметры “Слой нейронов” и “Слой кодирования” влияли менее существенно.

Опытным путем были определены оптимальные диапазоны параметров: уровни кодирования – 8...12, число скрытых нейронов – 80...500, число слоев – 6...10, число эпох – 20...400, размер пакета – 1024...2048. Кроме того, чем больше исходное изображение, тем более высокие значения перечисленных параметров требуются (за исключением числа эпох, которое можно ограничить во избежание переобучения).

Таблица 1 – Влияние настроек сети на loss и размер файла весов (размер пакета 2048)

Table 1 – Impact of network settings on loss and weight file size (batch size 2048)

Слой кодирования	Нейроны	Слой нейронов	Эпохи	Цвет. пр-во	Время обучения (с)	Loss	Размер модели (Кб)	Сжатие
12	100	8	20	LAB	1452,2	0,00106	140,3	111,2x
12	100	8	20	RGB	880,5	0,00113	140,4	111,2x
12	100	8	20	YCbCr	1026,2	0,00042	140,2	111,3x
2	100	8	20	YCbCr	893,7	0,00063	133,0	117,4x
10	10	8	20	YCbCr	307,9	0,00098	3,8	4083,2x
10	100	2	20	YCbCr	354,5	0,00067	28,4	549,8x
10	100	8	2	YCbCr	98,1	0,00060	138,2	113,0x

В результате тестов также было установлено, что использование цветового пространства YCbCr позволяет уменьшить значение функции потерь примерно в два раза по сравнению с RGB, при этом не искажает цвета восстановленного изображения и лишь немного замедляет обучение. После завершения обучения полученный файл весов сжимался архиватором, а затем проводилось восстановление изображения с помощью загруженных весов.

Для демонстрации результатов выбрано три модели с различными настройками параметров, которые приведены в Таблице 2. Графики и подробные результаты также отслеживались для каждой модели в процессе обучения (Рисунок 2).

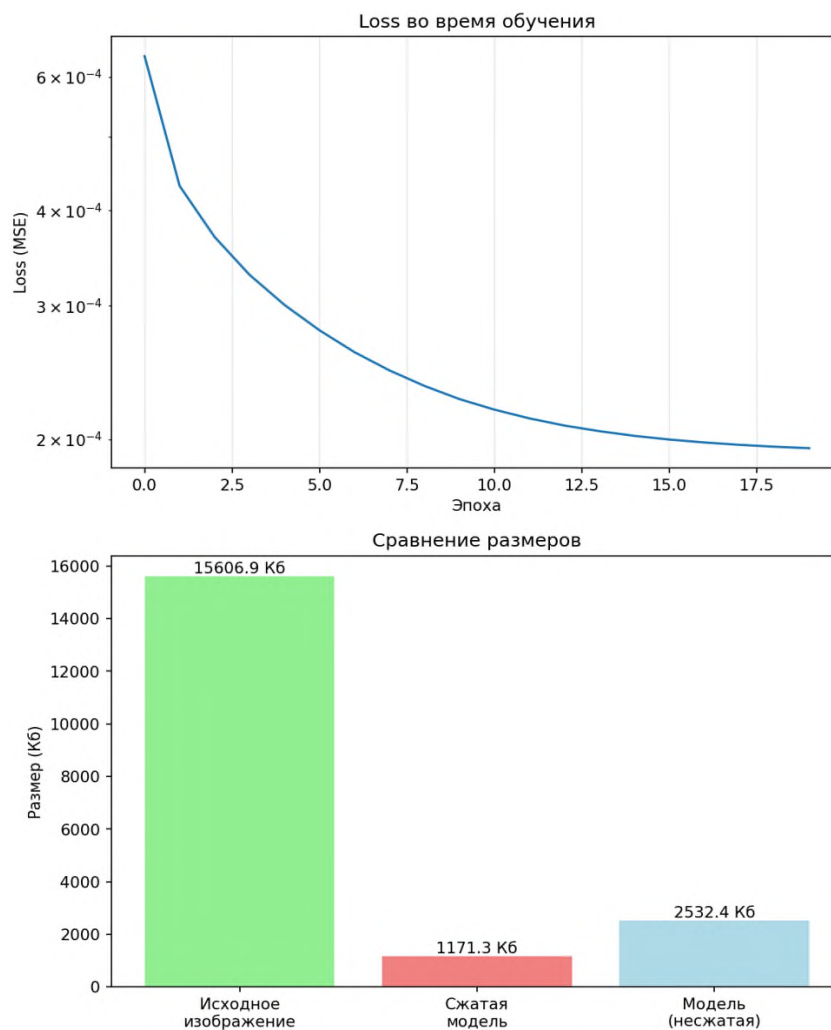


Рисунок 2 – Процесс обучения модели: снижение функции потерь (Loss) и сравнение размеров исходного изображения со сжатой моделью

Figure 2 – Model training process: loss function reduction and comparison of the original image size with the compressed model

Таблица 2 – Параметры моделей 1–3 и результаты (размер пакета 2048)  
Table 2 – Parameters of models 1–3 and results (batch size 2048)

Слои кодирования	Нейроны	Слои нейронов	Эпохи	Время обучения (с)	Loss	Размер модели (Кб)	Сжатие
12	200	8	36	3508,6	0,000264	530,0	29,4x
12	300	8	20	4746,4	0,000195	1171,3	13,3x
12	1000	8	36	55095,1	0,000063	12668,3	1,2x

Модель 1 обучалась при умеренном числе нейронов, что позволило достичь потерь порядка  $2,64 \cdot 10^{-4}$ .

Модель 2 имела большее число скрытых нейронов по сравнению с моделью 1. Это позволило снизить loss, но вес файла возрос до 1,17 Мб.

Модель 3 использовала наибольшее число нейронов, что дало приближенный к целевому loss ( $\approx 6,3 \cdot 10^{-5}$ ), но привело к сильному росту размера файла весов (около 12,7 Мб).

По отношению к исходному размеру изображения коэффициент сжатия существенно варьируется в зависимости от параметров модели и в рассмотренных экспериментах изменяется в диапазоне от 1,2 до 29,4 раза. Для всех моделей после завершения обучения файл весов нейронной сети дополнительно сжимался стандартным архиватором; при этом в каждом случае степень сжатия составляла примерно 2,2 раза. Анализ показал, что увеличение числа скрытых нейронов приводит к резкому росту времени обучения и размера файла весов при сравнительно небольшом снижении значения функции потерь. Так, при увеличении числа нейронов с 200 до 1000 время обучения возросло более чем в 15 раз, а размер модели увеличился на порядок, тогда как достигнутое улучшение качества восстановления (уменьшение loss до порядка  $6.3 \cdot 10^{-5}$ ) оказалось недостаточным с учетом целевого значения функции потерь порядка  $1 \cdot 10^{-5}$ .

Для ускорения процесса сжатия обучением, создана репродукция этого метода на основе фреймворка pytorch и включенной поддержкой CUDA. Ниже приведены результаты тестов этого решения. В процессе работы было использовано около 4 Гбайт видеопамати (RTX 3070 – 8 Gb) для размера пакета 65536. Оптимальными для этой версии программы стали параметры 24-300-6-36 (предпоследняя строка в Таблице 3), позволяющие сохранить приемлемое качество изображения и достигнуть большей степени сжатия<sup>2</sup>, при этом существенно снижая время обработки по сравнению с CPU версией программы.

Таблица 3 – Параметры и результаты тестирования моделей в pytorch+CUDA версии (размер пакета 65536)

Table 3 – Parameters and testing results for models in the pytorch+CUDA version (batch size 65536)

Слой кодирования	Нейроны	Слой нейронов	Эпохи	Время обучения (с)	Loss	Размер модели (Кб)	Сжатие
12	200	8	36	154,2	0,000515	598,5	26,1x
12	300	8	20	162,9	0,000498	1322,2	11,8x
12	1000	8	36	1461,3	0,000130	14424,5	1,1x
24	100	6	36	74,4	0,000602	130,0	120,1x
24	200	6	36	137,4	0,000413	475,5	32,8x
24	300	6	36	224,2	0,000308	1037,8	15,0x
24	300	8	36	288,2	0,000294	1360,3	11,5x

### Обсуждение

Полученные результаты показывают, что предложенный подход имеет свои сильные и слабые стороны. С одной стороны, нейросетевой способ хранения изображений действительно позволяет добиться очень высокого качества восстановления с низким loss, что дает потенциал для минимизации потерь при сжатии. К примеру, модель 1 с относительно небольшим весом сети (0,53 Мб) восстанавливала изображение с минимальными искажениями. С другой стороны, обучение таких моделей требует значительных вычислительных ресурсов и времени, особенно при увеличении числа параметров сети. Как видно из сравнений моделей 1–3, снижение loss достигается за счет резкого увеличения объема весов и времени обучения, что в целом делает метод нецелесообразным для практического использования в условиях ограниченных ресурсов. Кроме того, обычные алгоритмы сжатия при сопоставимом качестве работают за доли секунды на современных CPU/GPU, в отличие от предложенного метода на

<sup>2</sup> MexIvanov. *neuro-image-compression-pytorch*. GitHub. URL: <https://github.com/MexIvanov/neuro-image-compression-pytorch> (дата обращения: 22.02.2026).

изображении той же размерности. Хотя в данной работе использовалась архитектура MLP, современные исследования также комбинируют сверточные сети (CNN) и трансформеры для схожих задач [8]. Для повышения эффективности кодирования изображений полезным может быть использование дополнительной информации об углах и границах объектов [9]. В качестве перспективных моделей восстановления могут рассматриваться диффузионные архитектуры, показавшие эффективность в задачах улучшения качества изображения [10].

Тем не менее, у предложенного способа сжатия есть интересные перспективы. В первую очередь, он позволяет сохранить больше тонкой информации об изображении при сжатии (низкий loss) за счет времени выполнения. Во-вторых, метод может быть адаптирован под другие задачи. Например, в будущем нейросетевой подход можно рассмотреть для сжатия и хранения видеоданных: обучаемая последовательность кадров могла бы храниться в виде весов сети, с целью дальнейшего восстановления. Это могло бы дать преимущество в сохранении качества при длительной передаче или архивации видеоданных. Также исследование нейросетевых методов сжатия изображений может быть полезно для создания гибридных алгоритмов, в которых традиционные методы компрессии сочетаются с обучаемыми моделями для критичных к качеству участков изображения.

В перспективе методы нейросетевого сжатия могут интегрировать функции шумоподавления, аналогично подходу DeCompress [11]. Дальнейшее развитие также может касаться генеративного подхода к сжатию для улучшения качества восстановления изображений [12].

### Заключение

В данной работе исследован метод хранения изображений через обучение нейронной сети на одном примере, эффективно (математически) отображая его с помощью весов модели. Установлено, что нейронная сеть способна «запоминать» входное изображение настолько хорошо, что ее веса после обучения могут служить эффективным форматом сжатия изображения. При оптимальных параметрах сети восстановленное изображение по качеству максимально приближено к оригиналу (loss около  $10^{-5}$ ) при значительном сокращении объема данных в сжатом виде. Однако такой подход имеет потенциальный недостаток: длительное время обработки, по сравнению с классическими методами сжатия. Это время требуется для обучения модели на основе изображения и его затраты возрастают при необходимости получения восстановленного изображения большего качества. В сравнении с традиционными методами сжатия данная особенность делает подход нецелесообразным для практического использования в обычных условиях. Тем не менее, предложенный нейросетевой метод показал потенциал для достижения низких потерь при компрессии, и его можно рассматривать как перспективное направление исследований. Возможное продолжение работы включает оптимизацию архитектуры сети для снижения потерь при ускорении обучения, а также адаптацию подхода для сжатия видео или других типов данных, где потери качества критичны.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Подберезкин А.А., Остроух А.В., Борзенков А.М., Шмонин А.М., Пронин Ц.Б. Исследование комплексных подходов к цифровизации транспортных систем с применением методов искусственного интеллекта. *Транспорт и информационные технологии*. 2025;15(3):141–166. <https://doi.org/10.12731/3033-5965-2025-15-3-396>

- Podberezkin A.A., Ostroukh A.V., Borzenkov A.M., Shmonin A.M., Pronin C.B. Research methods for digitalization of transport systems using artificial intelligence. *Transportation and Information Technologies in Russia*. 2025;15(3):141–166. (In Russ.). <https://doi.org/10.12731/3033-5965-2025-15-3-396>
2. Archana R., Jeevaraj P.S.E. Deep learning models for digital image processing: a review. *Artificial Intelligence Review*. 2024;57(1). <https://doi.org/10.1007/s10462-023-10631-z>
  3. Jiang F., Tao W., Liu Sh., et al. An End-to-End Compression Framework Based on Convolutional Neural Networks. *IEEE Transactions on Circuits and Systems for Video Technology*. 2018;28(10):3007–3018. <https://doi.org/10.1109/TCSVT.2017.2734838>
  4. Sun H., Yu L., Katto J. *End-to-end learned image compression with quantized weights and activations*. arXiv. URL: <https://doi.org/10.48550/arXiv.2111.09348> [Accessed 22<sup>nd</sup> February 2026].
  5. Лёзин И.А., Соловьёв А.В. Сжатие изображений с использованием многослойного перцептрона. *Известия Самарского научного центра Российской академии наук*. 2016;18(4-4):770–773.  
 Lyozin I., Solovyov A. Performing an Image Compression by Using the Multilayer Perceptron. *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*. 2016;18(4-4):770–773. (In Russ.).
  6. Подберёзкин А.А., Борзенков А.М., Волков А.М., Пронин Ц.Б., Остроух А.В. Применение генетического алгоритма для оптимизации светофорного регулирования с учётом нормативных требований и приоритета общественного транспорта. *Современные наукоемкие технологии*. 2025;(9):181–185. <https://doi.org/10.17513/snt.40504>  
 Podberezkin A.A., Borzenkov A.M., Volkov A.M., Pronin Ts.B., Ostroukh A.V. Application of a Genetic Algorithm for Traffic Signal Optimization Considering Regulatory Requirements and Public Transport Priority. *Modern High Technologies*. 2025;(9):181–185. (In Russ.). <https://doi.org/10.17513/snt.40504>
  7. Deng H., Liu H., Wang F., Wang Zh., Wang Y. Image Compression Based on Genetic Algorithm and Deep Neural Network. In: *Bio-inspired Computing – Theories and Applications: 11<sup>th</sup> International Conference, BIC-TA 2016: Part I, 28–30 October 2016, Xi'an, China*. Singapore: Springer; 2016. P. 417–424. [https://doi.org/10.1007/978-981-10-3611-8\\_36](https://doi.org/10.1007/978-981-10-3611-8_36)
  8. Liu J., Sun H., Katto J. Learned image compression with mixed transformer-CNN architectures. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17–24 June 2023, Vancouver, BC, Canada*. IEEE; 2023. P. 14388–14397. <https://doi.org/10.1109/CVPR52729.2023.01383>
  9. Shindo T., Yamada K., Watanabe T., Watanabe H. Image coding for machines with edge information learning using segment anything. In: *2024 IEEE International Conference on Image Processing (ICIP), 27–30 October 2024, Abu Dhabi, United Arab Emirates*. IEEE; 2024. P. 3702–3708. <https://doi.org/10.1109/ICIP51287.2024.10647785>
  10. Li X., Ren Y., Jin X., et al. Diffusion models for image restoration and enhancement: A comprehensive survey. *International Journal of Computer Vision*. 2025;133(11):8078–8108. <https://doi.org/10.1007/s11263-025-02570-9>
  11. Zafari A., Chen X., Jalali Sh. DeCompress: Denoising via Neural Compression. In: *2025 IEEE International Symposium on Information Theory (ISIT), 22–27 June 2025, Ann Arbor, MI, USA*. IEEE; 2025. P. 1–6. <https://doi.org/10.1109/ISIT63088.2025.11195310>
  12. Nguyen N., Nguyen Th., Bose B. *Perception-based Image Denoising via Generative Compression*. arXiv. URL: <https://doi.org/10.48550/arXiv.2602.11553> [Accessed 22<sup>nd</sup> February 2026].

## ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Лоскутов Ярослав Дмитриевич**, студент, Московский автомобильно-дорожный государственный технический университет, Москва, Российская Федерация.

*e-mail:* [loskutoviaroslav@yandex.ru](mailto:loskutoviaroslav@yandex.ru)

ORCID: [0009-0006-9963-3271](https://orcid.org/0009-0006-9963-3271)

**Yaroslav D. Loskutov**, Student, Moscow Automobile and Road Construction State Technical University, Moscow, the Russian Federation.

**Грецкий Дмитрий Александрович**, студент, Московский автомобильно-дорожный государственный технический университет, Москва, Российская Федерация.

*e-mail:* [gr.mitya@gmail.com](mailto:gr.mitya@gmail.com)

ORCID: [0009-0002-0519-8397](https://orcid.org/0009-0002-0519-8397)

**Dmitry A. Gretskii**, Student, Moscow Automobile and Road Construction State Technical University, Moscow, the Russian Federation.

**Пронин Цезарь Борисович**, старший преподаватель, Московский автомобильно-дорожный государственный технический университет, Москва, Российская Федерация.

*e-mail:* [caesarpr12@gmail.com](mailto:caesarpr12@gmail.com)

ORCID: [0000-0002-9994-1032](https://orcid.org/0000-0002-9994-1032)

**Cesar B. Pronin**, Senior Lecturer, Moscow Automobile and Road Construction State Technical University, Moscow, the Russian Federation.

**Подберёзкин Александр Александрович**, старший преподаватель, Московский автомобильно-дорожный государственный технический университет, Москва, Российская Федерация.

*e-mail:* [a.podberezkin@gmail.com](mailto:a.podberezkin@gmail.com)

ORCID: [0009-0001-6192-5029](https://orcid.org/0009-0001-6192-5029)

**Alexander A. Podberezkin**, Senior Lecturer, Moscow Automobile and Road Construction State Technical University, Moscow, the Russian Federation.

**Остроух Андрей Владимирович**, доктор технических наук, профессор, Московский автомобильно-дорожный государственный технический университет, Москва, Российская Федерация.

*e-mail:* [ostroukh@mail.ru](mailto:ostroukh@mail.ru)

ORCID: [0000-0002-8887-6132](https://orcid.org/0000-0002-8887-6132)

**Andrey V. Ostroukh**, Doctor of Engineering Sciences, Professor, Moscow Automobile and Road Construction State Technical University, Moscow, the Russian Federation.

*Статья поступила в редакцию 24.02.2026; одобрена после рецензирования 23.03.2026; принята к публикации 27.03.2026.*

*The article was submitted 24.02.2026; approved after reviewing 23.03.2026; accepted for publication 27.03.2026.*