КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2026 Т. 18 № 1 С. 81–99
DOI: 10.20537/2076-7633-2026-18-1-81-99

Ки&М

МОДЕЛИ В ФИЗИКЕ И ТЕХНОЛОГИИ

УДК: 004.383.8.032.26

Нейроморфный процессор с аппаратным обучением на основе сверточной нейронной сети для анализа аудиоспектрограмм

М. О. Петров, Е. А. Рындин[a], Н. В. Андреева

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В. И. Ульянова (Ленина),
Россия, 197022, г. Санкт-Петербург, ул. Профессора Попова, д. 5

E-mail: [a] rynenator@gmail.com

Получено 20.09.2025, после доработки — 26.11.2025.
Принято к публикации 10.12.2025.

В статье предлагается архитектурное решение организации сверточной нейронной сети (СНС), ориентированное на аппаратную реализацию на конечных устройствах (edge-устройствах) в условиях ограниченных ресурсов. С этой целью предложен подход к сжатию спектрограмм до заданного размера (28×28) с использованием дискретизации, моноконверсии, оконного преобразования Фурье и двумерной интерполяции. Разработана сбалансированная процедура свертки на базе компактных сверточных фильтров, размер которых обеспечивает необходимый для edge-устройств баланс между вычислительной сложностью и точностью. Предложен алгоритм, позволяющий выполнять операции свертки и вычисления градиента функции ошибки на сверточном слое за один такт, обеспечивая повышение производительности режимов инференса и обучения СНС. Проведена оптимизация соотношения между обучаемостью сети и ее устойчивостью к переобучению за счет применения метода регуляризации Dropout с коэффициентом отбрасывания 0,5 для полносвязного слоя.

Работоспособность предложенного решения продемонстрирована на примере задачи распознавания аудиоспектрограмм звуков двигателей автомобилей и самолетов. СНС обучалась на сбалансированном наборе данных, состоящем из 7160 аудиозаписей. Обученная сеть демонстрировала высокую точность распознавания (95 %), низкие значения функции потерь ($< 0{,}2$), сбалансированные метрики «точность/полнота/F-мера», что свидетельствует об эффективности разработанной модели СНС.

Ключевые слова: нейроморфный процессор, аппаратный режим обучения, аудиоспектрограмма, сверточная нейронная сеть

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации — государственное задание в области научной деятельности FSEE-2025-0005.

COMPUTER RESEARCH AND MODELING
2026 VOL. 18 NO. 1 P. 81–99
DOI: 10.20537/2076-7633-2026-18-1-81-99

КиМ

MODELS IN PHYSICS AND TECHNOLOGY

UDC: 004.383.8.032.26

Neuromorphic processor with hardware learning based on a convolutional neural network for audio spectrogram analysis

M. O. Petrov, E. A. Ryndin[a], N. V. Andreeva

Saint Petersburg Electrotechnical University "LETI",
5 Professora Popova st., St. Petersburg, 197022, Russia

E-mail: [a] rynenator@gmail.com

Received 20.09.2025, after completion — 26.11.2025.
Accepted for publication 10.12.2025.

This paper proposes an architectural solution for organizing a convolutional neural network (CNN) oriented towards hardware implementation on edge devices under limited resources. To this goal, an approach to compressing spectrograms to a given size (28×28) is proposed using discretization, monoconversion, windowed Fourier transform, and two-dimensional interpolation. A balanced convolution procedure is developed based on compact convolutional filters, the size of which provides the balance between computational complexity and accuracy required for edge devices. An algorithm that enables convolution operations and calculation of the error function gradient in the convolutional layer in a single cycle ensuring increased performance in both inference and training modes of the CNN is proposed. The tradeoff between network trainability and its resistance to overfitting is optimized by applying the Dropout regularization method with a dropout coefficient of 0.5 for the fully connected layer.

The effectiveness of the proposed solution was demonstrated using the example of recognizing audio spectrograms of car and airplane engine sounds. The CNN was trained on a balanced dataset consisting of 7160 audio recordings. The trained network demonstrated high recognition accuracy (95 %), low loss values (< 0.2), and balanced precision/recall/F-metric, demonstrating the effectiveness of the developed CNN model.

Keywords: neuromorphic processor, hardware-assisted learning mode, audio spectrogram, convolutional neural network

Citation: *Computer Research and Modeling*, 2026, vol. 18, no. 1, pp. 81–99 (Russian).

The work was supported by the Ministry of Science and Higher Education of the Russian Federation — state assignment in the field of scientific activity FSEE-2025-0005.

Введение

В настоящее время алгоритмы формальных нейронных сетей активно используются и успешно решают многие прикладные задачи. Большая часть из них исполняется программно с помощью классических (фон-неймановских) вычислительных модулей, например графических (GPU) и тензорных (TPU) процессоров, а также с привлечением дополнительных ресурсов, таких как суперкомпьютеры, кластеры и облачные вычисления. Однако с увеличением сложности решаемых задач сильно увеличиваются энергопотребление и требования к производительности используемых вычислительных систем.

С целью эффективного решения проблем с энергопотреблением и требованиями к точности вычислительных систем при их автономной реализации на конечных устройствах значительные усилия направлены на разработку альтернативных концепций архитектур, ориентированных на память, в том числе нейроморфных. В общем случае можно выделить два класса подобных архитектурных решений, направленных на повышение энергоэффективности исполнения нейросетевых алгоритмов:

- нейропроцессоры на традиционной электронной компонентной базе (ЭКБ) [Нейропроцессоры для импульсных нейронных сетей, 2024], организованные на основе не-фон-неймановской архитектуры: TrueNorth (IBM) [DeBole et al., 2019], Loihi (Intel) [Davies et al., 2021], GrAI One и GrAI VIP (Франция) [Moreira et al., 2020], Innatera T1, ODIN, MorphIC, SPOON, ReckOn (Нидерланды) [Frenkel et al., 2021], DeepSouth, Akida (Австралия) [Vanarse et al., 2019], Spikey, BrainScaleS, BrainScaleS-2 (Германия) [Pehle et al., 2022], DynapSEL, DynapCNN, DynapSE2, Speck, Xylo (Швейцария) [Moradi et al., 2017], Tianji, Tianjic, TianjicX (Китай) [Zheng, Shi, 2023], SpiNNaker, SpinNaker-2 (Англия) [Höppner et al., 2021], Алтай (Россия) [Гришанов и др., 2020]; в архитектуре таких процессоров пытаются воссоздать принципы обработки информации в мозгу [Киселев, 2020; Киселев и др., 2025], такие как параллелизм и асинхронность, импульсный характер передачи информации, возможность реализации локального обучения, разреженность потоков данных;

- нейроморфные архитектуры на базе вычислений «в памяти» и «рядом с памятью», такие как ISAAC [Shafiee et al., 2016], Pipelayer [Song et al., 2017], RENO [Liu et al., 2015], PUMA [Ankit et al., 2019], NeuRRAM [Wan et al., 2022], STELLAR [Zhang et al., 2023] и ряд современных нейроморфных чипов на базе гетерогенных архитектур [Khwa et al., 2025; Wan et al., 2022; Huang et al., 2023; Wen et al., 2023; Wen et al., 2024a; Wen et al., 2024b; Lele et al., 2024; Chang et al., 2022; Hung et al., 2021; Hung et al., 2022; Spetalnick et al., 2022; Yoon et al., 2021; Hsu et al., 2023]. Основная идея, заложенная в данный аппаратный концепт, — это сокращение расходов на перемещение данных между процессором и памятью за счет реализации вычислений в аналоговом виде с использованием новых видов памяти, таких как резистивная память (RRAM), память на основе фазового перехода (PCM), магниторезистивная память (MRAM), встроенная флэш-память. В классических архитектурах блоки памяти и вычислительные блоки разделены в пространстве, что ограничивает эффективность вычислений и увеличивает энергопотребление из-за постоянной передачи данных между памятью и ЦПУ. «Вычисления в памяти» позволяют значительно повысить производительность и энергоэффективность аппаратно реализованных вычислительных архитектур за счет уменьшения количества операций, требуемых для аппаратного умножения матриц, по сравнению с тем количеством, которое требуется при использовании как графических ускорителей, так и нейросетевых (тензорных) процессоров [Sun et al., 2023], а также позволяют избавиться от проблемы «узкого горлышка» архитектуры фон Неймана классических вычислительных устройств. Подобный класс нейроморфных архитектур также предполагает максимальное распараллеливание данных и их асинхронную

обработку. Целевой сегмент их использования пока ограничен автономными edge-устройствами.

Резюмируя путь развития нейроморфных архитектур на базе вычислений в памяти, на примере использования RRAM, можно сказать, что изначально в основе принципа построения нейропроцессоров закладывалась преимущественно тайловая (от англ. *tile*) архитектура (за исключением процессора RENO). Тайл включает входные и выходные буферы; функциональный блок, аккумулирующий и объединяющий расчетные данные с блоков обработки данных; определенное количество блоков обработки данных, каждый из которых состоит из кроссбар-массива, АЦП, ЦАП, регистров, сдвига и суммирования. Матрично-векторное умножение выполняется в аналоговом виде на кроссбар-массивах с обвязкой из АЦП и ЦАП, тайлы объединяются в слои с общей КМОП-логикой. Аппаратное исполнение нейросетевых алгоритмов на данной архитектуре подразумевает конвейерный принцип обработки данных. Оптимизация тайловой архитектуры на начальном этапе развития происходила в направлении повышения ее гибкости, т. е. оптимизировалась глубина вычислительного конвейера, обеспечивающая его эффективную загрузку, совершенствовались способы разложения слоев нейронной сети по тайлам. С развитием технологии резистивной памяти произошел переход от этапа моделирования с использованием экспериментальных электрофизических характеристики мемристоров к этапу их аппаратной реализации. Как следствие, фокус развития нейроморфных архитектур сместился к разработке схем программирования ячеек мемристорных кроссбар-массивов, обеспечивающих необходимую точность вычислений в памяти; схем коррекции резистивного состояния ячеек для минимизации ошибок, обусловленных их «залипанием»; а также к совершенствованию способов организации матрично-векторного перемножения для снижения энергопотребления. На этом этапе возникли трудности в переносе режима обучения на аппаратную часть, обусловленные конечным числом резистивных уровней мемристора, а также особенностями совмещения аналоговых вычислений с цифровой логикой [Hong, Chung, 2024].

На основании анализа существующих решений нейроморфных чипов [Sheridan et al., 2017; Prezioso et al., 2015; Li et al., 2018; Yao et al., 2020; Cai et al., 2019; Wan et al., 2020; Liu et al., 2020; Xue et al., 2020; Yin et al., 2020] можно сделать заключение о том, что основное затруднение, возникающее при построении нейроморфных архитектур на базе аналоговых вычислений в памяти, связано с поиском оптимального баланса между производительностью, гибкостью архитектуры и точностью вычислений, ею обеспечиваемой.

В статье предлагается вариант архитектурного решения аналоговой асинхронной сверточной нейронной сети (СНС), предназначенной для аппаратной реализации режимов инференса и обучения в условиях ограниченных ресурсов (edge-устройства). Разработанная архитектура ориентирована на аппаратное исполнение на базе разработанных нами ранее пяти универсальных КМОП-IP-блоков [Petrov et al., 2024a; Petrov et al., 2024b] для аналоговой реализации всех вычислительных операций в режимах обучения и инференса. При этом кроссбар-массивы функциональных аналоговых КМОП-блоков с цифровым управлением уровнем проводимости обеспечивают выполнение операции матрично-векторного умножения в сверточном и полносвязном слоях без использования ЦАП и с применением АЦП в цепях управления весами синаптических связей только в режиме обучения [Рындин, Андреева, 2025]. Подобный подход к проектированию топологии СНС при повышении производительности обеспечивает значительное снижение энергопотребления нейронной сети как в инференс-режиме, так и в режиме обучения, благодаря отсутствию АЦП и ЦАП на входах и выходах аналоговых кроссбар-массивов и использованию для перестройки и хранения синаптических весов асинхронных цифровых КМОП-схем, не потребляющих мощность в стационарных состояниях.

Работоспособность разработанной архитектуры проверялась на решении задачи распознавания звуковых спектрограмм и определения источника звука с использованием базы дан-

ных записей звуков двигателей машин и самолетов из открытых источников в сети Интернет (https://zvukipro.com/transport).

Методы

Для оптимизации ресурсов, требуемых для аппаратной реализации СНС, матрица входов сети ограничивалась размером 28×28, приводя к необходимости разбиения звуковых записей на временные интервалы длительностью 300 мс и разработки способа сжатия исходных спектрограмм (размером $M_{orig} \times N_{orig} = 128 \times 10$) данных временных интервалов, не приводящего к потере точности.

С этой целью проводилась предобработка аудиозаписей программой, написанной на языке Python 3 с использованием библиотеки pydub [Robert, 2011] и реализующей следующий алгоритм:

- исходный непрерывный во времени t звуковой сигнал $x(t)$ дискретизируется:

$$x[n] = x(nT_s), \quad T_s = \frac{1}{f_s}, \quad n = 0, 1, \ldots, N-1, \tag{1}$$

где $x[n]$ — дискретный сигнал, n — номер отсчета дискретного сигнала, T_s, f_s — период и частота дискретизации соответственно;

- дискретизированный сигнал $x[n]$ разделяется на временные интервалы $x_m[l]$ длительностью по 300 мс (длина окна в отсчетах $N_m = 0{,}3f_s$):

$$x_m[l] = x[l + m \cdot N_m] \cdot w[l], \quad l = 0, 1, \ldots, N_m - 1, \tag{2}$$

где m — индекс окна, $w[l]$ — оконная функция Ханна;

- выполняется преобразование стереофонического сигнала в моносигнал с использованием выражения

$$x_m[l] = \frac{x_{m,left}[l] + x_{m,right}[l]}{2}, \tag{3}$$

где $x_{m,left}[l]$, $x_{m,right}[l]$ — отсчеты сигналов левого и правого стереоканалов соответственно;

- выполняется оконное преобразование Фурье (ОПФ) [Allen, Rabiner, 1977; Jeon et al., 2020], позволяющее изучать как частотный спектр, так и его изменение во времени. В отличие от обычного преобразования Фурье, которое показывает только глобальные частотные компоненты сигнала (т. е. совокупность частот за весь период записи без учета их временной локализации), ОПФ разбивает сигнал в интервале 300 мс на короткие временные окна (в данном проекте на 10 окон длительностью по 93 мс, покрывающих интервал 300 мс с наложениями по 69 мс между смежными окнами) и применяет преобразование Фурье к каждому из них отдельно:

$$X[m, k] = \sum_{l=0}^{N_m-1} x_m[l] \cdot e^{-\frac{j2\pi kl}{N_m}}, \quad k = 0, \ldots, \frac{N_m}{2}; \tag{4}$$

- с использованием оконного преобразования Фурье находится линейная спектрограмма:

$$S_{linear}[m, k] = 20 \log_{10} |(X[m, k])|, \quad k = 0, \ldots, \frac{N_m}{2}; \tag{5}$$

- формируется мел-спектрограмма размером 128×10 в соответствии с выражением

$$M[m, b] = \sum_{k=0}^{N_m/2} \left|(X[m, k])^2\right| \cdot \Phi_b[k], \quad k = 0, \ldots, \frac{N_m}{2}, \tag{6}$$

$$\Phi_b[k] = \begin{cases} 0, & f_k < f_{b-1}, \\ \dfrac{f_k - f_{b-1}}{f_b - f_{b-1}}, & f_{b-1} \leqslant f_k < f_b, \\ \dfrac{f_{b+1} - f_k}{f_{b+1} - f_b}, & f_b \leqslant f_k < f_{b+1}, \\ 0, & f_k \geqslant f_{b+1}, \end{cases}$$

где $\Phi_b[k]$ — функция взвешивания, определяющая вклад частотных компонент f_k в мел-полосы, f_{b-1}, f_b, f_{b+1} — минимальная, центральная и максимальная частоты мел-полосы с индексом b.

Следует отметить, что, в зависимости от уровня шума в исходном сигнале, длительность окон $x_m[l]$ может варьироваться. Увеличение длительности окон (при соответствующем увеличении общей длительности входного сигнала) и выполнение оконного преобразования Фурье (4) по большему числу отсчетов в каждом окне позволяют выделить информативные спектральные составляющие на фоне шумовых, обеспечивая необходимую точность классификации сигналов.

После предобработки производилось сжатие исходной спектрограммы S размером $M_{orig} \times$ $\times N_{orig}$ (128×10) к размеру 28×28 с использованием двумерной интерполяции с коэффициентом α по оси частот и коэффициентом β по временной оси:

$$S_{resized}[u, v] = S\left[\frac{\alpha}{u}, \frac{\beta}{v}\right], \quad u = 0, 1, \ldots, 27, \quad v = 0, 1, \ldots, 27, \tag{7}$$

где $S_{resized}[u, v]$ — спектрограмма целевого размера 28×28, u, v — индексы элементов спектрограммы.

Схема архитектуры СНС для решения задачи классификации звуковых спектрограмм представлена на рис. 1.

Сверточная нейронная сеть состоит из двух частей. Первая часть предназначена для реализации инференс-режима и включает (рис. 1):

- сверточный слой, включающий три фильтра K_r, $r = 1, 2, 3$, с размером ядра 4×4, входными данными для которого являются элементы спектрограммы S размером 28×28 пикселей с одним каналом (градации серого). В результате свертки формируется матрица карты признаков C_M с использованием следующего алгоритма:

  – преобразование входной спектрограммы S размером 28×28 в матрицу S_M размером 625×16 в соответствии с рис. 1, *б* и алгоритмом, который может быть описан следующим псевдокодом:

```
I_M = I_S − I_K + 1;
J_M = J_S − J_K + 1;
for i_M = 1 : I_M
    for j_M = 1 : J_M
        for i = 1 : I_K
            for j = 1 : J_K
```

$$S_M(I_M * (j_M - 1) + i_M,\ J_K * (i - 1) + j) = S(i + j_M - 1,\ j + i_M - 1);$$

```
            end
        end
    end
end
```

где I_S, J_S — число строк и столбцов входной спектрограммы; i, j — индексы строк и столбцов сверточного фильтра; I_K, J_K — число строк и столбцов сверточного фильтра; i_M, j_M — индексы строк и столбцов матрицы S_M; I_M, J_M — число строк и столбцов матрицы S_M;

– преобразование сверточных фильтров в массив K_M размером 16×3 построчным выстраиванием в r-й столбец 16×1 элементов фильтра K_r, $r = 1, 2, 3$ (рис. 1, *б*);

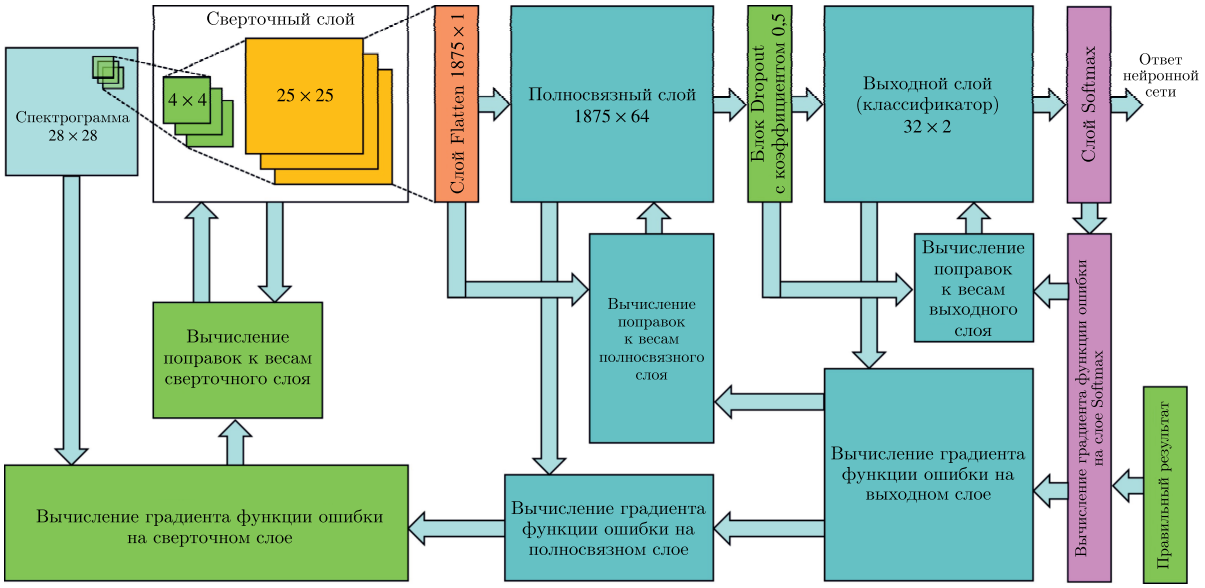
– вычисление матрицы признаков C_M размером 625×3 в результате матричного умножения (рис. 1, *б*):

$$C_M = S_M \times K_M. \tag{8}$$

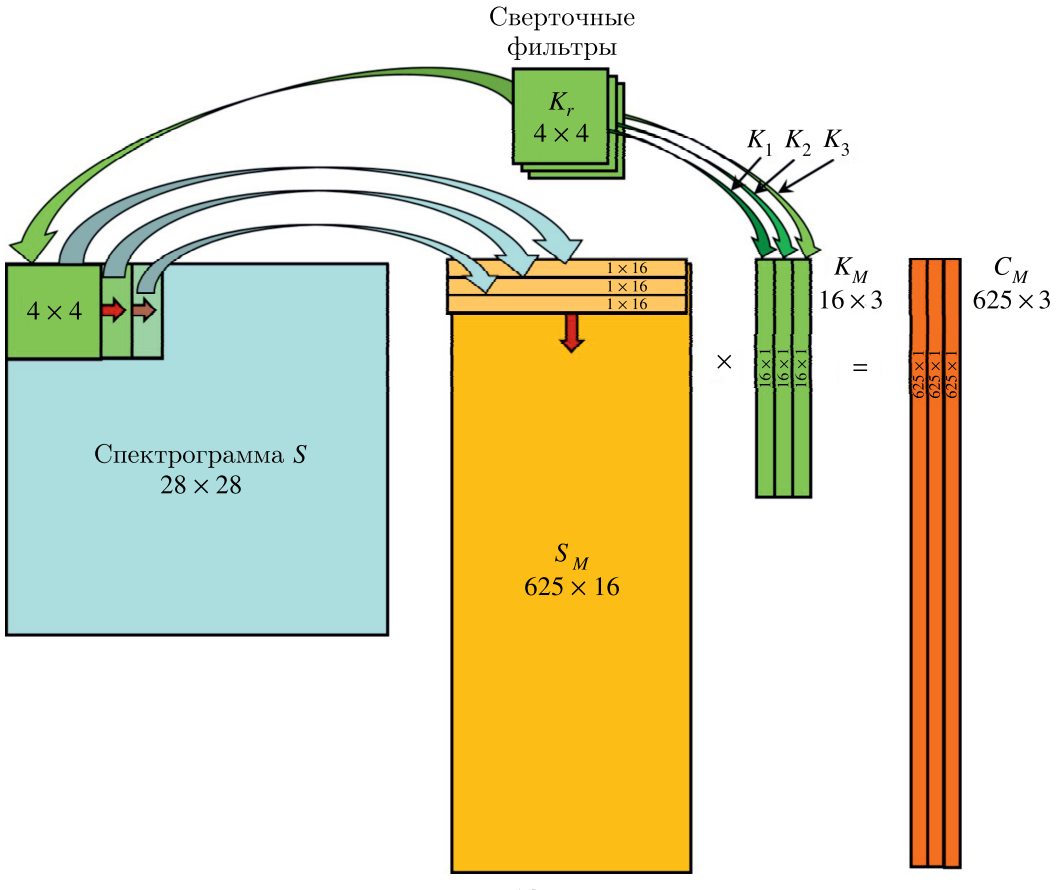
В предложенном алгоритме выполнения свертки преобразование входной спектрограммы S размером 28×28 в матрицу S_M размером 625×16 для параллельного умножения на весовые коэффициенты сверточных фильтров выполняется дублированием входных каналов в соответствии с приведенным выше псевдокодом и на аппаратном уровне не требует дополнительных операций и ресурсов. Алгоритм оптимизирован для аппаратной реализации матрично-векторного умножения в аналоговом домене с использованием либо новых видов памяти (например, мемристивных кроссбар-массивов), либо разработанных нами ранее аналоговых синаптических КМОП-элементов с цифровым хранением весов. Размер матрицы массива синаптических элементов (кроссбар-массива) при используемом подходе задается размером массива сверточных фильтров K_M, а их количество B определяется числом строк матрицы S_M. При этом количество матриц может варьироваться в пределах от $B = 1$ (при выполнении свертки за число тактов $T = (R_S - R_K + 1)^2$, где R_S, R_K — число строк (столбцов) в исходной аудиоспектрограмме S и в сверточном фильтре K_r, соответственно) до $B = T$ (при выполнении свертки за один такт), в зависимости от требований к производительности и энергопотреблению системы. Общее число операций умножения и суммирования в предложенном методе аппаратной реализации свертки не изменится по сравнению с традиционным подходом;

- слой Flatten, преобразующий двумерную карту признаков C_M, полученную после свертки, в одномерный вектор C_{1D} размером 1875×1 для передачи в полносвязный слой;

- полносвязный слой с 64 нейронами и функцией активации ReLU с матрицей весов W_L размером 1875×64, в котором для предотвращения переобучения используется метод прореживания (блок Dropout), деактивирующий случайным образом 50 % нейронов во время обучения, формируя выходной вектор данных D размером 32×1. Значение коэффициента Dropout, выбиралось на основе серии экспериментов, в которых сравнивались значения 0,3, 0,5 и 0,7. При значении 0,3 наблюдались признаки переобучения (растущий разрыв между точностью на обучающей и валидационной выборках), а при значении 0,7 — признаки недообучения (медленная сходимость и низкая точность на обеих выборках). Значение 0,5 позволило достичь оптимального баланса между стабильностью обучения и обобщающей способностью модели, что подтвердилось высокими и устойчивыми метриками на валидационной выборке;



(а)



(б)

Рис. 1. Архитектура сверточной нейронной сети (а) и иллюстрация выполнения операции свертки и формирования матрицы признаков C_M (б)

- выходной слой с матрицей весов W_{OUT} размером 32×2, выполняющий функцию классификатора на два класса;

- слой Softmax, выполняющий преобразование результата классификации $Y = \begin{bmatrix} y[1] \\ y[2] \end{bmatrix}$ в распределение вероятностей $P = \begin{bmatrix} p[1] \\ p[2] \end{bmatrix}$ в соответствии с выражением

$$p[r] = \frac{\exp(y[r])}{\sum_{h=1}^{2} \exp(y[h])}, \quad r = 1, 2, \tag{9}$$

где r — индекс класса, $p[r]$ — вероятность того, что спектрограмма соответствует классу r.

Распределение вероятностей P является выходным результатом СНС.

Вторая часть реализует режим обучения методом обратного распространения ошибки и включает (рис. 1):

- блок вычисления градиента функции ошибки G_S на слое Softmax как разности векторов вероятностного распределения P и правильной реакции сети F на предъявленное входное изображение:

$$G_S = P - F; \tag{10}$$

- блок вычисления матрицы поправок ΔW_{OUT} к весовым коэффициентам выходного слоя в соответствии с выражением

$$\Delta W_{OUT} = D \times G_S^T; \tag{11}$$

- блок вычисления градиента функции ошибки на выходном слое G_{OUT} как результат матрично-векторного умножения:

$$G_{OUT} = W_{OUT} \times G_S; \tag{12}$$

- блок вычисления матрицы поправок ΔW_L к весовым коэффициентам полносвязного слоя в соответствии с выражением

$$\Delta W_L = C_{1D} \times G_{OUT,D}^T, \tag{13}$$

где $G_{OUT,D}^T$ — транспонированный вектор градиента функции ошибки на выходном слое, в котором дополнительные элементы с индексами, соответствующими индексам инактивированных блоком Dropout столбцов полносвязного слоя, имеют нулевые значения;

- блок вычисления градиента функции ошибки на полносвязном слое G_L:

$$G_L = W_L \times G_{OUT,D}; \tag{14}$$

- блок вычисления градиента функции ошибки на сверточном слое в виде матрицы G_C размером 16×3:
  - преобразование вектора градиента функции ошибки на полносвязном слое G_L размером 1875×1 в матрицу G_{LM} размером 625×3;
  - вычисление градиентов функции ошибки на сверточном слое в результате матричного умножения:

$$G_C = S_M^T \times G_{LM}; \tag{15}$$

- блок вычисления поправок $\Delta k_{r,i,j}$ к весам сверточного слоя

$$K_r = \begin{bmatrix} k_{r,1,1} & k_{r,1,2} & k_{r,1,3} & k_{r,1,4} \\ k_{r,2,1} & k_{r,2,2} & k_{r,2,3} & k_{r,2,4} \\ k_{r,3,1} & k_{r,3,2} & k_{r,3,3} & k_{r,3,4} \\ k_{r,4,1} & k_{r,4,2} & k_{r,4,3} & k_{r,4,4} \end{bmatrix},$$

где $r = 1, 2, 3$, в соответствии с выражением

$$\Delta k_{r,i,j} = k_{r,i,j} - \gamma \times g_{C,r,I_K(j-1)+i}, \tag{16}$$

где $g_{C,r,I_K(j-1)+i}$ — элементы градиента функции ошибки на сверточном слое, γ — константа обучения (гиперпараметр нейронной сети).

Результаты

Для обучения СНС использовалась выборка из 7160 звуковых записей, сбалансированная по классам (то есть выборка, в которой классы представлены равномерно). Данные были разделены на обучающую, валидационную и тестовую выборки в количественном соотношении 80 % : 10 % : 10 %.

На рис. 2 представлены звуковые сигналы двигателя самолета (рис. 2, *а*) и мотора автомобиля (рис. 2, *б*), а также их линейные и мел-спектрограммы. Результаты преобразования спектрограмм приведены на рис. 3, *а* для самолета и на рис. 3, *б* для автомобиля.

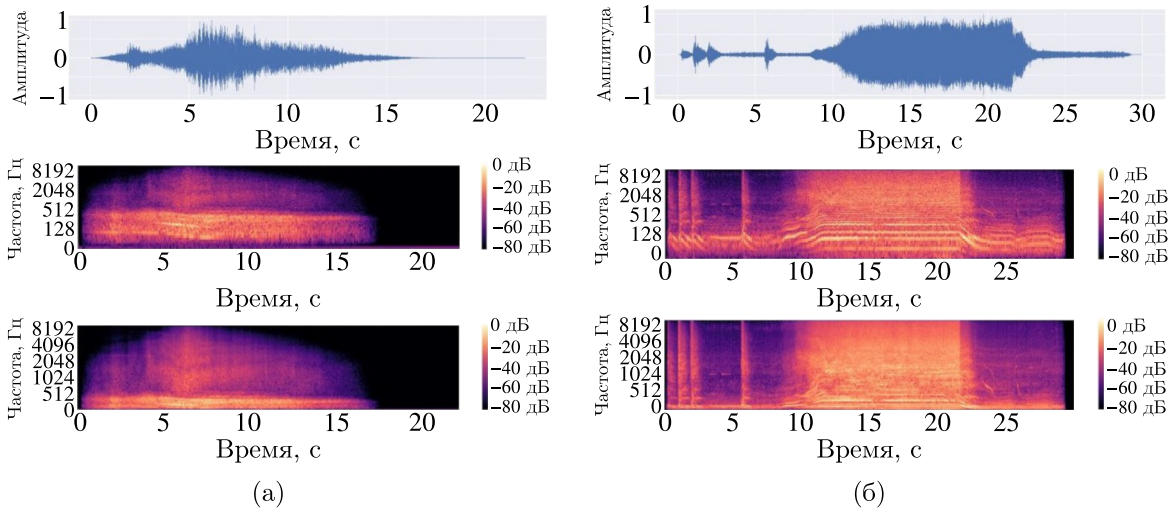


Рис. 2. Звуковой сигнал самолета, его линейная и мел-спектрограмма (а) и звуковой сигнал мотора автомобиля и, соответственно, его линейная и мел-спектрограмма (б). Каждая из приведенных спектрограмм размером 128 × 1000 представляет собой совокупность спектрограмм размером 128 × 10, полученных для 100 временных интервалов исходных аудиосигналов длительностью по 300 мс

На рис. 4 приведены количественные показатели обученной сверточной нейронной сети для классификации звуковых спектрограмм. В течение первых 10–15 эпох наблюдаются быстрый рост точности на валидационной выборке и резкое снижение функции потерь. После примерно 30 эпох обе метрики стабилизируются, что свидетельствует о достижении оптимального баланса между обучением и обобщением. Финальные показатели, достигнутые на валидационной выборке:

- точность (Accuracy) ≈ 95 %;
- значения функции потерь (Loss) < 0,2.

Оценки энергоэффективности разработанной архитектуры СНС, выполненные для варианта ее интегральной реализации на основе синаптических КМОП-элементов, предложенных нами в работах [Petrov et al., 2024a; Рындин, Андреева, 2025], приведены в табл. 1 без учета энергозатрат на предобработку аудиозаписей. Средний ток синаптического КМОП-элемента в режиме

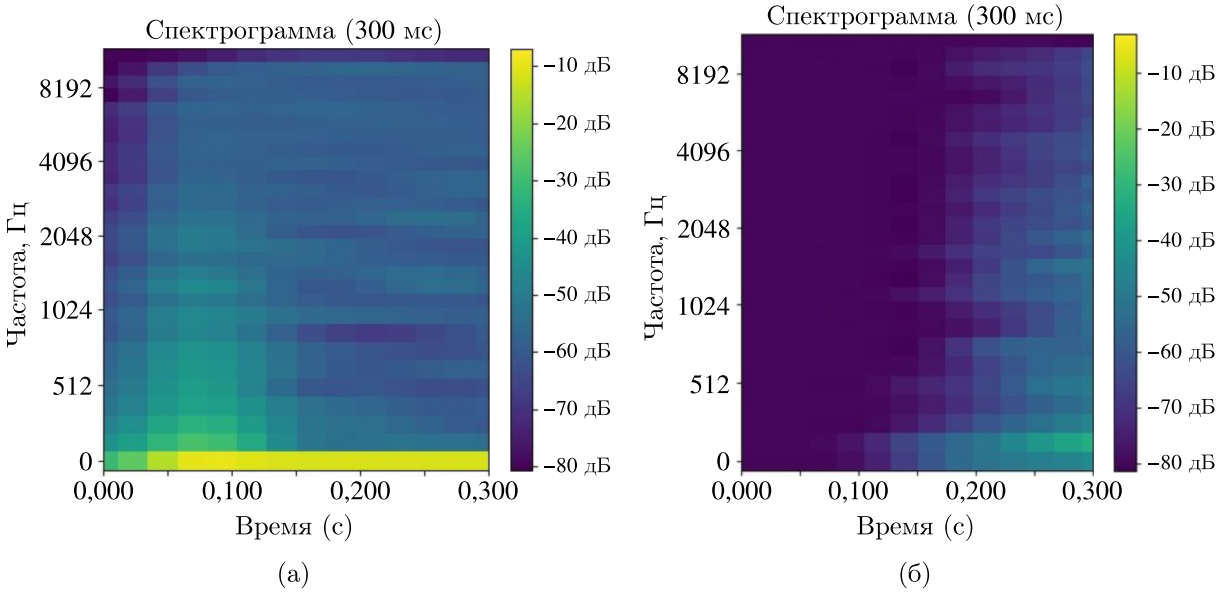


Рис. 3. Результаты преобразования исходной спектрограммы размером 128×10 к размеру 28×28 для самолета (а) и для автомобиля (б)

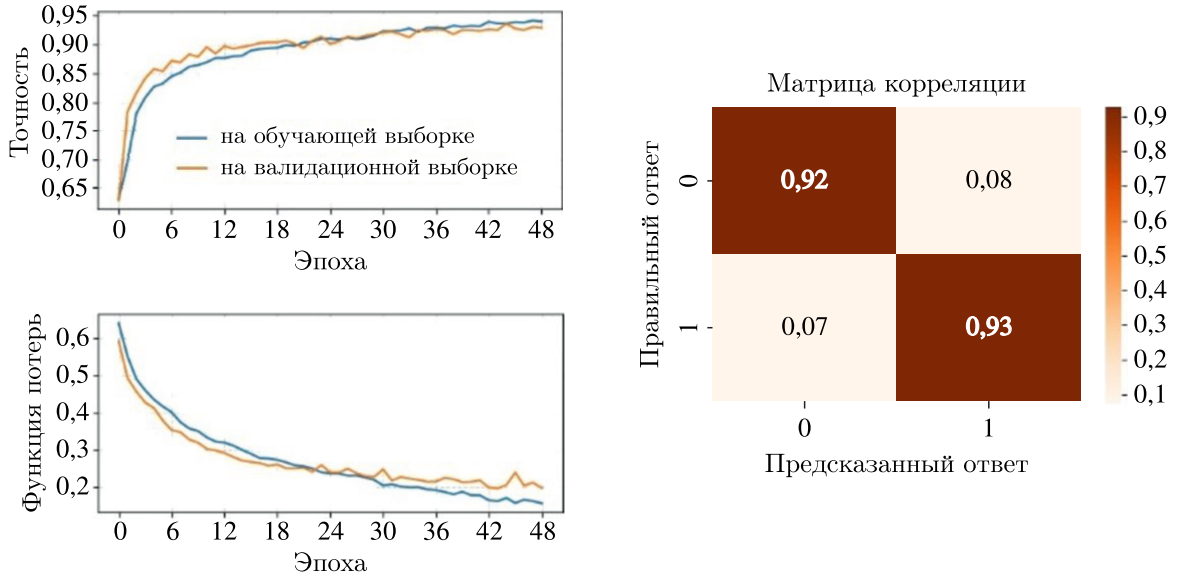


Рис. 4. Результат обучения нейронной сети

обучения I_{AV} и длительность временного интервала перестройки синаптического веса t_W были определены в результате SPICE-моделирования схемы элемента, экстрагированной из его топологии с учетом параметров транзисторов и соединений.

Приведенные в табл. 1 оценки средней мощности, потребляемой СНС в режимах обучения (P_{TR}) и инференса (P_{INF}), а также энергоэффективности F (Оп/Дж) в режиме обучения, были получены с использованием следующих выражений:

$$P_{TR} = V_{DD} I_{AV} t_W N_{TR} V_{TR}, \tag{17}$$

$$P_{INF} = V_{DD} I_{AV} t_W N_{INF} V_{INF}, \tag{18}$$

$$F = (V_{DD} I_{AV} t_W)^{-1}, \tag{19}$$

Таблица 1. Результаты оценки энергоэффективности предложенной архитектуры СНС

Параметр	Значение	Единицы измерения
Технология	КМОП	—
Проектная норма	50	нм
Напряжение питания V_{DD}	0,7	В
Средний ток синаптического элемента в режиме обучения I_{AV}	390	мкА
Длительность временного интервала перестройки синаптического веса и выполнения аналогового умножения t_W	320	пс
Средняя мощность, потребляемая синаптическим КМОП-элементом в режиме обучения $P_{AV} = V_{DD} \cdot I_{AV}$	273	мкВт
Средняя энергия перестройки синаптического веса и операции аналогового умножения $A_{AV} = P_{AV} \cdot t_W$	87,36	фДж/Оп
Число операций, выполняемых для обработки одной аудиоспектрограммы (асп.) в режиме инференса N_{INF}	150 000	Оп/асп.
Число операций, выполняемых для обработки одной аудиоспектрограммы в режиме обучения N_{TR}	300 000	Оп/асп.
Энергия, затрачиваемая на обработку одной аудиоспектрограммы в режиме инференса $E_{INF} = N_{INF} \cdot A_{AV}$	13,1	нДж
Энергия, затрачиваемая на обработку одной аудиоспектрограммы в режиме обучения $E_{TR} = N_{TR} \cdot A_{AV}$	26,2	нДж
Потребляемая мощность при скорости обработки данных в режиме инференса $V_{INF} = 3 \cdot 10^3$ асп./с	40	мкВт
Потребляемая мощность при скорости обработки данных в режиме обучения $V_{TR} = 3 \cdot 10^3$ асп./с	80	мкВт
Энергоэффективность в режиме обучения $F = A_{AV}^{-1}$	11,44	ТОп/Дж

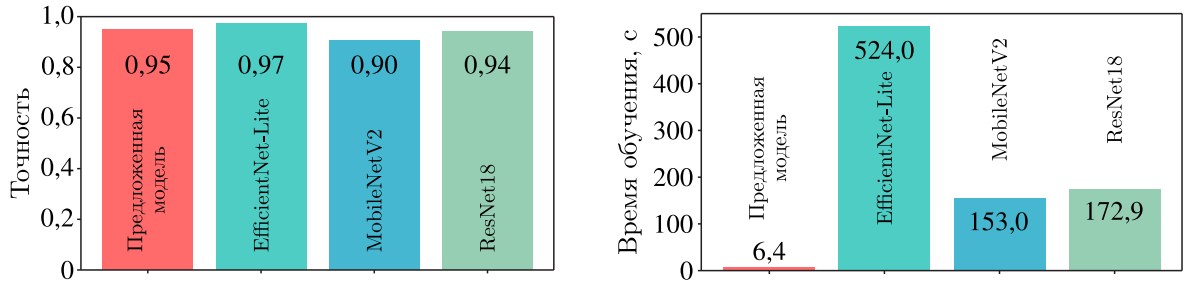


Рис. 5. Результаты сравнения точности (а) и времени обучения (б) предложенной модели СНС с известными решениями для нейросетевой аудиообработки

где V_{DD} — напряжение питания; N_{TR}, N_{INF} — число операций, выполняемых для обработки одной аудиоспектрограммы в режимах обучения и инференса соответственно; V_{TR}, V_{INF} — число аудиоспектрограмм, обрабатываемых в секунду, в режимах обучения и инференса соответственно.

Результаты сравнения предложенной архитектуры СНС с известными решениями для нейросетевой аудиообработки, такими как MobileNet V2, ResNet-18, MCUNet, EfficientNet-Lite, приведены на рис. 5, 6 и в табл. 2 [Chiang, Marculescu, 2024; Sandler, Howard, 2018; Al-Gaashani et al., 2025; Shahriar, 2025; Somvanshi et al., 2025; Lin et al., 2020; Liu, 2020].

Результаты сравнительного анализа разработанного архитектурного решения с известными решениями для нейросетевой аудиообработки свидетельствуют о том, что

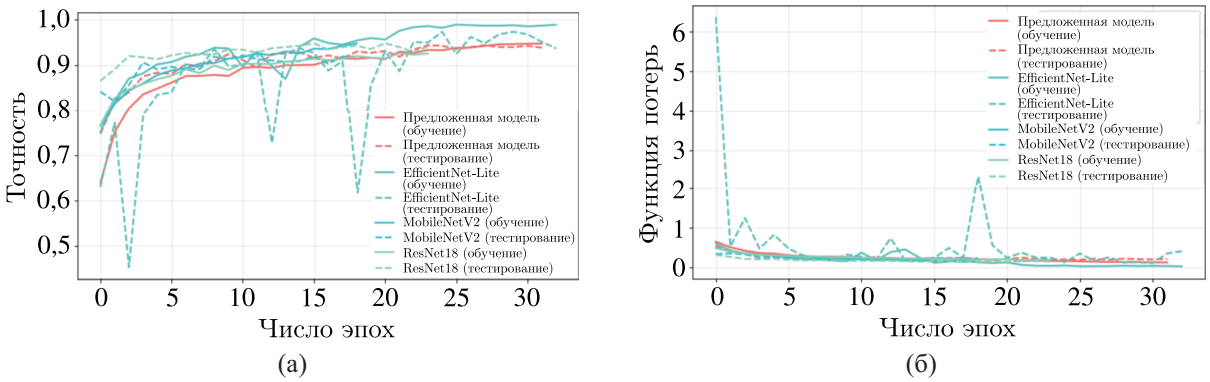


Рис. 6. Результаты сравнения зависимостей точности (а) и функции потерь (б) от числа эпох обучения предложенной модели СНС с известными решениями для нейросетевой аудиообработки

Таблица 2. Результаты сравнения предложенной архитектуры СНС с известными решениями для нейросетевой аудиообработки

Архитектура	Предложенная модель	MobileNet V2 [Sandler, Howard, 2018]	ResNet-18 [Shahriar, 2025; Somvanshi et al., 2025]	EfficientNet-Lite [Liu, 2020]	MCUNet [Lin et al., 2020]	SCAN-Edge [Chiang, Marculescu, 2024]
Точность, база данных	~ 95 %, аудио «звуки двигателей»	~ 90 %, аудио «звуки двигателей»	~ 94 %, аудио «звуки двигателей»	~ 98 %, аудио «звуки двигателей»	~ 71 %, ImageNet 1K, 512 KB SRAM	76–78 %, ImageNet-1K
Параметры, млн	~ 0,1	~ 3,4	~ 11,7	~ 5,3	< 1	3–5
Сложность, GMACs / MOPS	< 10 MOPS	~ 0,3 GMACs	~ 1,8 GMACs	~ 0,39 GMACs	~ 70 MOPS	~ 0,4 GMACs
Обучение на чипе	Да	Нет	Нет	Нет	Нет	Нет
Аппаратная платформа	ПЛИС / ASIC	ЦПУ, ГПУ, Edge TPU	ЦПУ, ГПУ МК* STM32 (в версии TinyML)	ЦПУ, ГПУ	МК* (Cortex-M4/M7)	Edge ЦПУ / ГПУ
Потребляемая мощность, мВт	< 1	~ 250	~ 500	~ 15–20	< 2	~ 200

* МК — микроконтроллер.

- предобработка входных звуковых сигналов с использованием дискретизации, моноконверсии, оконного преобразования Фурье и двумерной интерполяции для формирования спектрограмм уменьшенного размера обеспечивает минимизацию требуемых аппаратных ресурсов для реализации СНС и снижение энергопотребления, что важно для вычислений конечными устройствами;

- использование компактных сверточных фильтров (размер которых 4×4) в соответствии с представленными результатами обучения (рис. 4) обеспечивает необходимый для edge-устройств баланс между вычислительной сложностью и точностью СНС;

- преобразование входной спектрограммы S размером 28×28 в матрицу S_M размером 625×16, трех сверточных фильтров K_r, $r = 1, 2, 3$, размером 4×4 в массив K_M размером 16×3 и вектора градиента функции ошибки на полносвязном слое G_L размером 1875×1 в матрицу G_{LM} размером 625×3 позволяет, в зависимости от требований к производительности и энергопотреблению системы, а также к используемым аппаратным ресурсам, выполнять операции свертки и вычисления градиента функции ошибки на сверточном слое за меньшее число тактов по сравнению с традиционным подходом (в пределе — за один такт), обеспечивая оптимизацию производительности и энергопотребления системы в режимах инференса и обучения;

- применение для полносвязного слоя метода регуляризации Dropout с коэффициентом отбрасывания 0,5, значение которого используется в качестве гиперпараметра, обеспечивает оптимизацию соотношения между обучаемостью СНС и ее устойчивостью к переобучению, делают модель СНС эффективной для задач классификации аудиоспектрограмм конечными устройствами с ограниченными ресурсами.

Заключение

В работе предложена модель сверточной нейронной сети для классификации аудиоспектрограмм звуков двигателей автомобилей и самолетов с аппаратной реализацией режима обучения, предусматривающая предобработку и эффективное сжатие исходных аудиоспектрограмм к размеру 28×28 пикселей, включающая сверточный слой с тремя фильтрами размером 4×4, полносвязный слой с 64 нейронами с функцией активации ReLU и использованием функции Dropout с коэффициентом 0,5 для предотвращения переобучения, выходной слой классификатора и набор функциональных блоков, реализующих аппаратное обучение методом обратного распространения ошибки.

Предложенная модель ориентирована на аппаратное исполнение с использованием разработанного нами ранее подхода к построению топологии глубоких нейронных сетей с обучением на чипе [Petrov et al., 2024b], основанного на аналоговой реализации всех вычислительных операций, включая матрично-векторное умножение в синаптических кроссбар-массивах, при цифровой реализации операций настройки и хранения синаптических весов. Так, базовый блок сети с обучением на кристалле (ASIC-дизайн) реализуется на основе пяти унифицированных аналоговых КМОП-IP-блоков [Petrov et al., 2024a]:

- для инференс-режима сети:
  - универсальный элемент для сверточного и полносвязного слоев, в котором каждый вход данного элемента имеет связь с каждым выходом, при этом связь представляет собой перестраиваемый весовой коэффициент;
  - элемент, выполняющий функцию Softmax;
- для реализации обучения:
  - элемент, осуществляющий вычисление разности двух векторов;
  - элемент для умножения каждого из входов на значение на последнем входе;
  - элемент для нормирования получаемых промежуточных величин (поддержания всех численных результатов в определенном диапазоне значений).

Результаты обучения СНС на сбалансированном наборе данных из 7160 аудиозаписей показывают, что разработанное архитектурное решение обучается и эффективно решает бинарную задачу классификации звуков двигателей автомобилей и самолетов с высокой точностью (95 %) и малым значением функции потерь (< 0,2), обладает сбалансированными метриками «точность/полнота/F-мера» и, таким образом, потенциально подходит для реализации аппаратного исполнения режимов инференса и обучения рассматриваемого типа нейронных сетей в условиях ограниченных ресурсов на автономных конечных устройствах благодаря низкой вычислительной сложности и высокой точности классификации.

Список литературы (References)

Гришанов Н. В., Зверев А. В., Ипатов Д. Е., Канглер В. М., Катомин М. Н., Коденко Н. И., Кострицын И. А., Макаров Ю. С., Мамычев В. И., Павлов П. В., Панченко К. Е., Полстянкин А. В. Нейроморфный процессор «Алтай» для энергоэффективных вычислений // Наноиндустрия. — 2020. — № S96-2. — С. 531–538.

Grishanov N. V., Zverev A. V., Ipatov D. E., Kangler V. M., Katomin M. N., Kodenko N. I., Kostriysyn I. A., Makarov Yu. S., Mamychev V. I., Pavlov P. V., Panchenko K. E., Polstyankin A. V. Neiromorfnyi processor “Altai” dlya energoeffektivnykh vychisleniy [Neuromorphic processor Altai for energy-efficient computing] // Nanoindustriya. — 2020. — No. S96-2. — P. 531–538 (in Russian).

Киселев М. В. Исследование двухнейронных ячеек памяти в импульсных нейронных сетях // Компьютерные исследования и моделирование. — 2020. — Т. 12, № 2. — С. 401–416.

Kiselev M. V. Issledovanie dvukhneironnykh yacheek pamyati v impul’snykh neironnykh setyakh [Exploration of 2-neuron memory units in spiking neural networks] // Computer Research and Modeling. — 2020. — Vol. 12, No. 2. — P. 401–416 (in Russian).

Киселев М. В., Урусов А. М., Иваницкий А. Ю. Метод адаптивных гауссовых рецептивных полей для спайкового кодирования числовых переменных // Компьютерные исследования и моделирование. — 2025. — Т. 17, № 3. — С. 389–400.

Kiselev M. V., Urusov A. M., Ivanitsky A. Yu. Metod adaptivnykh gaussovykh receptivnykh polei dlya spaikovogo kodirovaniya chislovykh peremennykh [The adaptive Gaussian receptive fields for spiking encoding of numeric variables] // Computer Research and Modeling. — 2025. — Vol. 17, No. 3. — P. 389–400 (in Russian).

Нейропроцессоры для импульсных нейронных сетей // Нейроэлектроника и нейротехнологии будущего: Труды I Школы-конференции с международным участием. — Нижний Новгород, 2024.

Neiroprocessory dlya impul’snykh neironnykh setei [Neuroprocessors for pulse neural networks]. Trudy I Shkoly-konferencii s mezhdunarodnym uchastiem “Neiroelektronika i neirotekhnologii budushchego” [Proc. 1st School-Conf. with Int. Participation “Neuroelectronics and Neurotechnologies of the Future”]. — Nizhny Novgorod, 2024 (in Russian).

Рындин Е. А., Андреева Н. В. Интегральный электронный синаптический КМОП-элемент // Патент РФ № 2836650. — 2025.

Ryndin E. A., Andreeva N. V. Integral’nyi elektronnyi sinapticheskiy KMOP-element [Integrated electronic synaptic CMOS element] // Patent RU 2836650. — 2025 (in Russian).

Al-Gaashani M. S. A. M., Xu W., Obsie E. Y. MobileNetV2-based deep learning architecture with progressive transfer learning for accurate Monkeypox detection // Appl. Soft Comput. — 2025. — Vol. 169. — 112553.

Allen J. B., Rabiner L. R. A unified approach to short-time Fourier analysis and synthesis // Proc. of the IEEE. — 1977. — Vol. 65, No. 11. — P. 1558–1564.

Ankit A., Hajj I. E., Chalamalasetti S. R., Ndu G., Foltin M., Williams R. S., Faraboschi P., Hwu W., Strachan J. P., Roy K., Milojicic D. S. PUMA: a programmable ultra-efficient memristor-based accelerator for machine learning inference // Proc. of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. Providence, RI, USA, 13–17 April 2019. — New York, USA: ACM, 2019. — P. 715–731.

Cai F., Correll J. M., Lee S. H., Lim Y., Bothra V., Zhang Z., Flynn M. P., Lu W. D. A fully integrated reprogrammable memristor – CMOS system for efficient multiply – accumulate operations // Nat. Electron. — 2019. — Vol. 2, No. 7. — P. 290–299.

Chang M., Spetalnick S. D., Crafton B., Khwa W.-S., Chih Y.-D., Chang M.-F., Raychowdhury A. A 40nm 60.64TOPS/W ECC-capable compute-in-memory/digital 2.25MB/768KB RRAM/SRAM system with embedded cortex M3 microprocessor for edge recommendation systems // Proc. of the 2022 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 20–26 February 2022. — New York, USA: IEEE, 2022. — P. 1–3.

Chiang H.-Y., Marculescu D. SCAN-Edge: finding MobileNet-speed hybrid networks for diverse edge devices via hardware-aware evolutionary search // arXiv preprint. — 2024. — arXiv:2408.15395v1

Davies M., Wild A., Orchard G., Sandamirskaya Y., Guerra G. A. F., Joshi P., Plank P., Risbud S. R. Advancing neuromorphic computing with loihi: A survey of results and outlook // Proc. of the IEEE. — 2021. — Vol. 109, No. 5. — P. 911–934.

DeBole M. V., Taba B., Amir A., Akopyan F., Andreopoulos A., Risk W. P., Kusnitz J., Ortega Otero C., Nayak T. K., Appuswamy R., Carlson P. J., Cassidy A. S., Datta P., Esser S. K., Garreau G. J., Holland K. L., Lekuch S., Mastro M., McKinstry J., di Nolfo C., Paulovicks B., Sawada J., Schleupen K., Shaw B. G., Klamo J. L., Flickner M. D., Arthur J. V., Modha D. S. TrueNorth: Accelerating from zero to 64 million neurons in 10 years // Computer. — 2019. — Vol. 52, No. 5. — P. 20–29.

Frenkel C., Bol D., Indiveri G. Bottom-up and top-down approaches for the design of neuromorphic processing systems: Tradeoffs and synergies between natural and artificial intelligence // Proc. of the IEEE. — 2021. — Vol. 111, No. 6. — P. 623–652.

Hong S., Chung Y. C. CRPIM: An efficient compute-reuse scheme for ReRAM-based Processing-in-Memory DNN accelerators // J. Syst. Architect. — 2024. — Vol. 153. — 103192.

Höppner S., Yan Y., Dixius A., Scholze S., Partzsch J., Stolba M., Kelber F., Vogginger B., Neumärker F., Ellguth G., Hartmann S., Schiefer S., Hocker T., Walter D., Liu G., Garside J., Furber S., Mayr C. The SpiNNaker 2 processing element architecture for hybrid digital neuromorphic computing // arXiv preprint. — 2021. — arXiv:2103.08392

Hsu H.-H., Wen T.-H., Huang W.-H., Khwa W.-S., Lo Y.-C., Jhang C.-J., Chin Y.-H., Chen Y.-C., Lo C.-C., Liu R.-S., Tang K.-T., Hsieh C.-C., Chih Y.-D., Chang T.-Y. J., Chang M.-F. A nonvolatile AI-edge processor with SLC – MLC hybrid ReRAM compute-in-memory macro using current – voltage-hybrid readout scheme // IEEE J. Solid-State Circuits. — 2023. — Vol. 59, No. 1. — P. 116–127.

Huang W.-H., Wen T.-H., Hung J.-M., Khwa W.-S., Lo Y.-C., Jhang C.-J., Hsu H.-H., Chin Y.-H., Chen Y.-C., Lo C.-C., Liu R.-S., Tang K.-T., Hsieh C.-C., Chih Y.-D., Chang T.-Y., Chang M.-F. A nonvolatile AI-edge processor with 4MB SLC-MLC hybrid-mode ReRAM compute-in-memory macro and 51.4-251 TOPS/W // Proc. of the 2023 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 19–23 February 2023. — New York, USA: IEEE, 2023. — P. 15–17.

Hung J.-M., Wen T.-H., Huang Y.-H., Huang S.-P., Chang F.-C., Su C.-I., Khwa W.-S., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Chih Y.-D., Chang T.-Y. J., Chang M.-F. 8-b precision 8-Mb ReRAM compute-in-memory macro using direct-current-free time-domain readout scheme for AI edge devices // IEEE J. Solid-State Circuits. — 2022. — Vol. 58, No. 1. — P. 303–315.

Hung J.-M., Xue C.-X., Kao H.-Y., Huang Y.-H., Chang F.-C., Huang S.-P., Liu T.-W., Jhang C.-J., Su C.-I, Khwa W.-S., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Ho M.-S., Chou C.-C., Chih Y.-D., Chang T.-Y. J., Chang M.-F. A four-megabit compute-in-memory macro with eight-bit precision based on CMOS and resistive random-access memory for AI edge devices // Nat. Electron. — 2021. — Vol. 4, No. 12. — P. 921–930.

Jeon H., Jung Y., Lee S., Jung Y. Area-efficient short-time fourier transform processor for time – frequency analysis of non-stationary signals // Appl. Sci. — 2020. — Vol. 10, No. 20. — 7208.

Khwa W.-S., Wen T.-H., Hsu H.-H., Huang W.-H., Chang Y.-C., Chiu T.-C., Ke Z.-E., Chin Y.-H., Wen H.-J., Hsu W.-T., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Ho M.-S., Lele A. S., Teng S.-H., Chou C.-C., Chih Y.-D., Chang T.-Y. J., Chang M.-F. A mixed-precision memristor and SRAM compute-in-memory AI processor // Nature. — 2025. — Vol. 639. — P. 617–623.

Lele A. S., Chang M., Spetalnick S. D., Crafton B., Konno S., Wan Z., Bhat A., Khwa W.-S., Chih Y.-D., Chang M.-F., Raychowdhury A. A heterogeneous RRAM in-memory and SRAM near-memory SoC for fused frame and event-based target identification and tracking // IEEE J. Solid-State Circuits. — 2024. — Vol. 59, No. 1. — P. 52–64.

Li C., Hu M., Li Y., Jiang H., Ge N., Montgomery E., Zhang J., Song W., Dávila N., Graves C. E., Li Z., Strachan J. P., Lin P., Wang Z., Barnell M., Wu Q., Williams R. S., Yang J. J., Xia Q. Analogue signal and image processing with large memristor crossbars // Nat. Electron. — 2018. — Vol. 1, No. 1. — P. 52–59.

Lin J., Chen W.-M., Lin Y., Cohn J., Gan C., Han S. MCUNet: tiny deep learning on IoT devices // arXiv preprint. — 2020. — arXiv:2007.10319

Liu Q., Gao B., Yao P., Wu D., Chen J., Pang Y., Zhang W., Liao Y., Xue C.-X., Chen W.-H., Tang J., Wang Y., Chang M.-F., Qian H., Wu H. A fully integrated analog ReRAM based 78.4TOPS/W compute-in-memory chip with fully parallel MAC computing // Proc. of the 2020 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 16–20 February 2020. — New York, USA: IEEE, 2020. — P. 500–502.

Liu R. Higher accuracy on vision models with EfficientNet-Lite. — [Electronic resource]. — 2020. — https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html (accessed: 19.09.2025).

Liu X., Mao M., Liu B., Li H., Chen Y., Li B., Wang Y., Jiang H., Barnell M., Wu Q., Yang J. RENO: A high-efficient reconfigurable neuromorphic computing accelerator design // Proc. of the 52nd Annual Design Automation Conference 2015. San Francisco, CA, USA, 7–11 June 2015. — New York, USA: ACM, 2015. — P. 1–6

Moradi S., Qiao N., Stefanini F., Indiveri G. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs) // IEEE Trans. Biomed. Circuits Syst. — 2017. — Vol. 12, No. 1. — P. 106–122.

Moreira O., Yousefzadeh A., Chersi F., Kapoor A., Zwartenkot R.-J., Qiao P., Cinserin G., Khoei M. A., Lindwer M., Tapson J. NeuronFlow: A hybrid neuromorphic – dataflow processor architecture for AI workloads // Proc. of the 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS). Genova, Italy, 31 August 2020 – 02 September 2020. — New York, USA: IEEE, 2020. — P. 1–5.

Pehle C., Billaudelle S., Cramer B., Kaiser J., Schreiber K., Stradmann Y., Weis J., Leibfried A., Müller E., Schemmel J. The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity // Front. Neurosci. — 2022. — Vol. 16. — 795876.

Petrov M. O., Ryndin E. A., Andreeva N. V. Automated design of deep neural networks with in-situ training architecture based on analog functional blocks // The European Physical Journal Special Topics. — 2024a. — P. 1–14.

Petrov M. O., Ryndin E. A., Andreeva N. V. Compiler for hardware design of convolutional neural networks with supervised learning based on neuromorphic electronic blocks // Proc. of the 2024 Sixth International Conference Neurotechnologies and Neurointerfaces (CNN). Kaliningrad, Russian Federation, 19–21 September 2024. — New York, USA: IEEE, 2024b. — P. 1–4.

Prezioso M., Merrikh-Bayat F., Hoskins B. D., Adam G. C., Likharev K. K., Strukov D. B. Training and operation of an integrated neuromorphic network based on metal-oxide memristors // Nature. — 2015. — Vol. 521, No. 7550. — P. 61–64.

Robert J. Pydub. — [Electronic resource]. — 2011. — https://github.com/jiaaro/pydub (accessed: 19.09.2025).

Sandler M., Howard A. MobileNetV2: The next generation of on-device computer vision networks. — [Electronic resource]. — 2018. — https://research.google/blog/mobilenetv2-the-next-generation-of-on-device-computer-vision-networks/ (accessed: 19.09.2025).

Shafiee A., Nag A., Muralimanohar N., Balasubramonian R., Strachan J. P., Hu M., Williams R. S., Srikumar V. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars // ACM SIGARCH Computer Architecture News. — 2016. — Vol. 44, No. 3. — P. 14–26.

Shahriar T. Comparative analysis of lightweight deep learning models for memory-constrained devices // arXiv preprint. — 2025. — arXiv:2505.03303v1

Sheridan P. M., Cai F., Du C., Ma W., Zhang Z., Lu W. D. Sparse coding with memristor networks // Nat. Nanotechnol. — 2017. — Vol. 12, No. 8. — P. 784–789.

Somvanshi S., Islam M. M., Chhetri G., Chakraborty R., Mimi M. S., Shuvo S. A., Islam K. S., Javed S. A., Rafat S. A., Dutta A., Das S. From tiny machine learning to tiny deep learning: a survey // arXiv preprint. — 2025. — arXiv:2506.18927v1

Song L., Qian X., Li H., Chen Y. PipeLayer: a pipelined ReRAM-based accelerator for deep learning // Proc. of the 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA). Austin, TX, USA, 04–08 February 2017. — New York, USA: IEEE, 2017. — P. 541–552.

Spetalnick S. D., Chang M., Crafton B., Khwa W.-S., Chih Y.-D., Chang M.-F., Raychowdhury A. A 40nm 64kb 26.56TOPS/W 2.37Mb/mm2 RRAM binary/compute-in-memory macro with 4.23x improvement in density and > 75 % use of sensing dynamic range // Proc. of the 2022 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 20–26 February 2022. — New York, USA: IEEE, 2022. — P. 1–3.

Sun Z., Kvatinsky S., Si X., Mehonic A., Cai Y., Huang R. A full spectrum of computing-in-memory technologies // Nat. Electron. — 2023. — Vol. 6, No. 11. — P. 823–835.

Vanarse A., Osseiran A., Rassau A., van der Made P. A hardware-deployable neuromorphic solution for encoding and classification of electronic nose data // Sensors. — 2019. — Vol. 19, No. 22. — 4831.

Wan W., Kubendran R., Eryilmaz S. B., Zhang W., Liao Y., Wu D., Deiss S., Gao B., Raina P., Joshi S., Wu H., Cauwenberghs G., Wong H.-S. P. A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models // Proc. of the 2020 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 16–20 February 2020. — New York, USA: IEEE, 2020. — P. 498–500.

Wan W., Kubendran R., Schaefer C., Eryilmaz S. B., Zhang W., Wu D., Deiss S., Raina P., Qian H., Gao B., Joshi S., Wu H., Wong H.-S. P., Cauwenberghs G. A compute-in-memory chip based on resistive random-access memory // Nature. — 2022. — Vol. 608, No. 7923. — P. 504–512.

Wen T.-H., Hsu H.-H., Khwa W.-S., Huang W.-H., Ke Z.-E., Chin Y.-H., Wen H.-J., Chang Y.-C., Hsu W.-T., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Teng S.-H., Chou C.-C., Chih Y.-D., Chang T.-Y. J., Chang M.-F. A 22nm 16Mb floating-point ReRAM compute-in-memory macro with 31.2TFLOPS/W for AI edge devices‘// Proc. of the 2024 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 18–22 February 2024. — New York, USA: IEEE, 2024a. — P. 580–582.

Wen T.-H., Hung J.-M., Hsu H.-H., Wu Y., Chang F.-C., Li C.-Y., Chien C.-H., Su C.-I., Khwa W.-S., Wu J.-J., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Ho M.-S., Chih Y.-D., Chang T.-Y. J., Chang M.-F. A 28nm nonvolatile AI edge processor using 4Mb analog-based near-memory-compute ReRAM with 27.2 TOPS/W for tiny AI edge devices // Proc. of the 2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits). Kyoto, Japan, 11–16 June 2023. — New York, USA: IEEE, 2023. — P. 1–2.

Wen T.-H., Hung J.-M., Huang W.-H., Jhang C.-J., Lo Y.-C., Hsu H.-H., Ke Z.-E., Chen Y.-C., Chin Y.-H., Su C.-I, Khwa W.-S., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Ho M.-S., Chou C.-C., Chih Y.-D., Chang T.-Y. J., Chang M.-F. Fusion of memristor and digital compute-in-memory processing for energy-efficient edge computing // Science. — 2024b. — Vol. 384, No. 6693. — P. 325–332.

Xue C.-X., Huang T.-Y., Liu J.-S., Chang T.-W., Kao H.-Y., Wang J.-H., Liu T.-W., Wei S.-Y., Huang S.-P., Wei W.-C., Chen Y.-R., Hsu T.-H., Chen Y.-K., Lo Y.-C., Wen T.-H., Lo C.-C., Liu R.-S., Hsieh C.-C., Tang K.-T., Chang M.-F. A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices // Proc. of the 2020 IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, USA, 16–20 February 2020. — New York, USA: IEEE, 2020. — P. 244–246.

Yao P., Wu H., Gao B., Tang J., Zhang Q., Zhang W., Yang J. J., Qian H. Fully hardware-implemented memristor convolutional neural network // Nature. — 2020. — Vol. 577, No. 7792. — P. 641–646.

Yin S., Sun X., Yu S., Seo J. S. High-throughput in-memory computing for binary deep neural networks with monolithically integrated RRAM and 90-nm CMOS // IEEE Trans. Electron Devices. — 2020. — Vol. 67, No. 10. — P. 4185–4192.

Yoon J.-H., Chang M., Khwa W.-S., Chih Y.-D., Chang M.-F., Raychowdhury A. A 40nm 100Kb 118.44TOPS/W ternary-weight compute-in-memory RRAM macro with voltage-sensing read and write verification for reliable multi-bit RRAM operation // Proc. of the 2021 IEEE Custom Integrated Circuits Conference (CICC). Austin, TX, USA, 25–30 April 2021. — New York, USA: IEEE, 2021. — P. 1–2.

Zhang W., Yao P., Gao B., Liu Q., Wu D., Zhang Q., Li Y., Qin Q., Li J., Zhu Z., Cai Y., Wu D., Tang J., Qian H., Wang Y., Wu H. Edge learning using a fully integrated neuro-inspired memristor chip // Science. — 2023. — Vol. 381, No. 6663. — P. 1205–1211.

Zheng H., Shi L. Coherence in intelligent systems // Proc. of the 16th International Conference “Artificial General Intelligence” (AGI 2023). Stockholm, Sweden, 16–19 June 2023. — Cham, Switzerland: Springer, 2023. — P.‘357–366.