

Информатика и вычислительная техника Informatics and computer engineering

Научная статья

УДК 004.93'12

DOI: 10.14529/ctcr250301

ПРИМЕНЕНИЕ НЕЙРОСЕТЕВЫХ АЛГОРИТМОВ ОБРАБОТКИ ИНФОРМАЦИИ В АВТОНОМНЫХ СИСТЕМАХ ВИДЕОМОНИТОРИНГА

М.Г. Вахитов^{1,2}, vakhitovmg@susu.ru, <https://orcid.org/0000-0002-1286-3888>

А.В. Портнов², portnovav@susu.ru, <https://orcid.org/0009-0002-2649-219X>

А.Н. Николаев², nikolaevan@susu.ru, <https://orcid.org/0009-0002-9466-8847>

¹ АО «Челябинский радиозавод «Полет», Челябинск, Россия

² Южно-Уральский государственный университет, Челябинск, Россия

Аннотация. Статья посвящена проблеме обработки видеoinформации с использованием нейросетей в автономных системах в условиях ограниченных вычислительных ресурсов. **Цель исследования.** Целью статьи является исследование алгоритмов обработки видеoinформации в автономных системах с ограниченными вычислительными ресурсами на основе двухэтапной обработки данных. На первом этапе проводится определение контуров наблюдаемых объектов с помощью традиционных методов выделения ключевых признаков. Второй этап включает обработку с использованием нейросетевых алгоритмов, нацеленных на классификацию и распознавание объектов. Благодаря первичной обработке видеопотока существенно снижается объем информации, обрабатываемой нейронной сетью. **Материалы и методы.** В статье предложен вариант алгоритма первичной обработки видеoinформации, основанный на методах определения контуров объектов. Методы выделения границ, основанные на пространственной фильтрации, могут применяться как к стационарным, так и к подвижным объектам. Вторичная обработка основана на применении сверточной нейронной сети, коэффициенты которой подвергались квантованию для снижения вычислительной нагрузки. Для обучения нейронной сети были сгенерированы два набора данных (датасеты) для обучения, валидации и тестирования. Первый датасет состоял из исходных изображений с каким-либо объектом, второй содержал идеальный контур объектов (маски). Проведено сравнение двух нейронных сетей, обученных на исходных изображениях и на контурных масках изображений. Рассмотрены особенности реализации нейронных сетей на ПЛИС и предложена структурная схема проведения вычислений. Показана возможность итерационных вычислений с повторным использованием вычислительных ресурсов на каждой итерации (слое нейронной сети). **Результаты.** Приведены результаты предварительных исследований по обучению нейронной сети и квантованию ее коэффициентов, рассмотрены особенности аппаратной реализации нейронных сетей на базе программируемых логических интегральных схем. **Заключение.** Двухэтапный подход к обработке видеoinформации демонстрирует высокую эффективность в автономных системах, где существуют ограничения по энергопотреблению и вычислительным ресурсам. Статья может быть полезна разработчикам автономных систем видеомониторинга, обнаружения и сопровождения объектов наземного и воздушного базирования.

Ключевые слова: системы видеомониторинга, ПЛИС, нейронные сети, классификация изображений, обработка видеoinформации, первичная обработка видеoinформации

Благодарности. Исследование выполнено за счет гранта Российского научного фонда № 25-29-20236, <https://rscf.ru/project/25-29-20236/>.

Для цитирования: Вахитов М.Г., Портнов А.В., Николаев А.Н. Применение нейросетевых алгоритмов обработки информации в автономных системах видеомониторинга // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2025. Т. 25, № 3. С. 5–15. DOI: 10.14529/ctcr250301

APPLICATION OF NEURAL NETWORK ALGORITHMS INFORMATION PROCESSING IN AUTONOMOUS VIDEO MONITORING SYSTEMS

M.G. Vakhitov^{1,2}, vakhitovmg@susu.ru, <https://orcid.org/0000-0002-1286-3888>

A.V. Portnov², portnovav@susu.ru, <https://orcid.org/0009-0002-2649-219X>

A.N. Nikolaev², nikolaevan@susu.ru, <https://orcid.org/0009-0002-9466-8847>

¹ JSC "Chelyabinsk Radio Plant "Polyot", Chelyabinsk, Russia

² South Ural State University, Chelyabinsk, Russia

Abstract. The article is devoted to the problem of processing video information using neural networks in autonomous systems with limited computing resources. **The purpose of the study.** The purpose of the article is to study algorithms for processing video information in autonomous systems with limited computing resources based on two-stage data processing. At the first stage, the contours of the observed objects are determined using traditional methods of identifying key features. The second stage involves processing using neural network algorithms aimed at classifying and recognizing objects. Due to the primary processing of the video stream, the amount of information processed by the neural network is significantly reduced. **Materials and methods.** The article offers a variant of the algorithm for primary processing of video information based on methods for determining the contours of objects. Boundary allocation methods based on spatial filtering can be applied to both stationary and mobile objects. Secondary processing is based on the use of a convolutional neural network, the coefficients of which were quantized to reduce the computational load. To train the neural network, two datasets were generated for training, navigation, and testing. The first dataset consisted of initial images with an object, the second contained an ideal contour of objects (masks). A comparison of two neural networks trained on the source images and on contour masks of the images is carried out. The features of the implementation of neural networks on FPGAs are considered and a block diagram of the calculations is proposed. The possibility of iterative calculations with repeated use of computing resources at each iteration (neural network layer) is shown. **Results.** The results of preliminary studies on neural network training and quantization of its coefficients are presented, and the specifics of hardware implementation of neural networks based on programmable logic integrated circuits are considered. **Conclusion.** The two-stage approach to video information processing demonstrates high efficiency in autonomous systems where there are limitations on energy consumption and computing resources. The article may be useful to developers of autonomous video monitoring systems, detection and tracking of ground-based and air-based facilities.

Keywords: video monitoring systems, FPGA, neural networks, image classification, video information processing, primary processing of video information

Acknowledgments. The study was supported by the Russian Science Foundation grant No. 25-29-20236, <https://rscf.ru/project/25-29-20236/>.

For citation: Vakhitov M.G., Portnov A.V., Nikolaev A.N. Application of neural network algorithms information processing in autonomous video monitoring systems. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2025;25(3):5–15. (In Russ.) DOI: 10.14529/ctcr250301

Введение

В настоящее время все более стремительно происходит развитие автономных систем обработки видеoinформации. Такие системы могут применяться в беспилотных транспортных средствах, беспилотных летательных аппаратах, умных системах видеонаблюдения и др. Во всех случаях обработка видеoinформации должна производиться в режиме реального времени. В отличие от стационарных высокопроизводительных решений, автономные устройства не обладают большими вычислительными ресурсами, так как в основном работают на относительно маломощных процессорах, микроконтроллерах или программируемых логических интегральных схемах (ПЛИС). Это формирует особые требования к алгоритмам обработки видеoinформации. Они должны быть нацелены на обеспечение высокой скорости и эффективности анализа видеопотока.

Современные методы обработки видеoinформации, основанные на применении нейронных сетей, демонстрируют хорошие результаты в обнаружении, классификации, распознавании объектов и т. д. Но их применение в автономных системах часто невозможно из-за отсутствия необходимого количества ресурсов. В качестве решения данной проблемы может послужить разделение обработки информации на этапы – первичную (классические методы обработки) и вторичную обработку (обработка с помощью нейронных сетей). Данный принцип проектирования систем позволит значительно упростить структуру используемой нейронной сети ввиду предварительной обработки входных данных.

Первичная обработка основана на традиционных методах выделения ключевых признаков объекта в кадре (границ, факта перемещения, направления движения, формы и т. д.). Для этого могут применяться алгоритмы пространственной фильтрации, которые не требуют больших вычислительных ресурсов. На данном этапе входные данные упрощаются, а их объем уменьшается. Вторичная обработка нацелена на интерпретацию результата первичной обработки с помощью более сложных алгоритмов – нейронных сетей.

Важным является выбор методов обработки для каждого этапа. Первичная обработка может быть основана на применении градиентных методов выделения границ. Несмотря на свою простоту и схожесть, все методы имеют свои преимущества и недостатки [1, 2]. Вторичная обработка предполагает использование более простых нейронных сетей, что позволит реализовать их на вычислительных платформах с ограниченными вычислительными ресурсами, в том числе на базе ПЛИС.

В настоящей статье рассмотрены алгоритмы первичной обработки видеoinформации с целью выделения контуров объектов, обсуждаются особенности аппаратной реализации нейронной сети на ПЛИС, приводятся сравнительные результаты обучения для предлагаемой двухэтапной обработки видеoinформации и традиционного подхода с непосредственным обучением нейронной сети на наборах видеоданных.

1. Алгоритмы первичной обработки видеoinформации

Выделение контуров является одним из главных этапов в системах видеомониторинга. Для неподвижных объектов чаще всего используются классические методы, основанные на пространственной фильтрации. Наиболее часто используются градиентные методы, такие как операторы Робертса, Прюитта, Собеля, Кирша, детектор границ Кэнни и др. [3–7]. Данные методы могут применяться и для подвижных объектов, классическим методом анализа которых является вычитание соседних кадров. При этом большее количество ложных границ, определенных каким-либо оператором, исчезнут. Также анализ последовательности кадров позволяет вычислять векторы перемещения объектов между кадрами, что необходимо для определения траекторий и оценки скорости движения объектов.

В общем случае использование градиентных методов нацелено на определение изменений интенсивности пикселей изображения. Процесс пространственной фильтрации можно представить в виде [3]

$$g(x, y) = \sum_{s=-\frac{m-1}{2}}^{\frac{m-1}{2}} \sum_{t=-\frac{n-1}{2}}^{\frac{n-1}{2}} w(s, t) f(x + s, y + t),$$

где m и n – размеры маски фильтра; $w(s, t)$ – коэффициенты маски; $f(x, y)$ – значения пикселей в окрестности выбранной точки.

Выбор типа оператора зависит от задачи, которая решается с помощью данного алгоритма. Главными отличиями являются различное качество детектирования границ и нагрузка на вычислительную систему, что особенно актуально для автономных систем. С одной стороны, методы с простыми коэффициентами (например, оператор Прюитта) обеспечивают минимальную вычислительную нагрузку, но обладают меньшей точностью. С другой – методы с коэффициентами, кратными степени двойки (оператор Собеля), обладают компромиссом между скоростью вычислений и качеством.

Сложностью при использовании градиентных методов является определение порогового значения, необходимого для отнесения конкретного пикселя к фону или объекту. На величину порога (область оптимальных значений) влияет как само изображение, так и тип исполь-

зуемой маски фильтра [8]. Для его определения могут применяться различные автоматизированные методы – метод Оцу, итеративного среднего, адаптивный порог [3] и др. Часто применяется и экспериментальный метод, который предполагает ручной подбор порогового значения.

Современные подходы часто объединяют традиционные методы с глубоким обучением (сверточные, рекуррентные нейронные сети и др.). Такой подход улучшает точность работы систем в сложных условиях, но требует значительных вычислительных ресурсов, вследствие чего увеличивается время обработки. В автономных системах реального времени необходимо соблюдать баланс между производительностью и объемом затрачиваемых ресурсов.

2. Особенности реализации нейронных сетей на ПЛИС

На сегодняшний день существует большое количество программных решений для реализации нейронных сетей в вычислительных системах на базе микропроцессоров и микроконтроллеров [9–11]. Однако в большинстве случаев такие решения не могут обеспечить требуемую производительность нейросети для обработки потока видеоинформации в реальном режиме времени, так как все вычисления в микропроцессорах выполняются последовательно.

Вместе с этим предлагаются аппаратные решения по реализации нейронных сетей с использованием ПЛИС [11, 12]. Такой подход позволяет существенно повысить производительность вычислений за счет возможности параллельной обработки информации, однако в силу специфики разработки программного обеспечения ПЛИС широкого распространения он не получил.

Для реализации нейронной сети на ПЛИС необходимо, в первую очередь, провести оценку вычислительных ресурсов. В качестве базового вычислительного узла можно принять один нейрон сети. Структура нейрона практически полностью совпадает со структурой цифрового фильтра с конечной импульсной характеристикой (КИХ) за исключением наличия на выходе нелинейной (в большинстве случаев) функции активации [13]. Современные ПЛИС имеют в своем составе аппаратные DSP-блоки, позволяющие эффективно реализовывать КИХ-фильтры. В качестве примера в табл. 1 приведена информация по DSP-блокам в доступных на рынке ПЛИС семейства CycloneV фирмы Intel [14].

Аппаратные DSP-блоки ПЛИС семейства CycloneV

Таблица 1

Hardware DSP blocks of the CycloneV FPGA family

Table 1

Обозначение семейства ПЛИС	Код	Количество умножителей при различной организации разрядности		
		9 × 9	18 × 18	27 × 27
Cyclone V E	A2	75	50	25
	A4	198	132	66
	A5	450	300	150
	A7	468	312	156
	A9	1026	684	342

Учитывая то, что ПЛИС может работать на тактовых частотах, превышающих 100 МГц, один DSP-блок позволяет проводить вычисления для одного нейрона в режиме реального времени. Число весовых коэффициентов в нейроне может достигать нескольких десятков и будет зависеть от темпа поступления видеоинформации (количества кадров в секунду) и тактовой частоты ПЛИС.

Вычисление функции активации нейрона может быть реализовано табличным способом с использованием встроенных блоков памяти ПЛИС. Кроме этого, встроенная память ПЛИС используется для хранения весовых коэффициентов обученной нейронной сети. В табл. 2 приведен пример доступных ресурсов памяти для ПЛИС семейства CycloneV фирмы Intel.

Таблица 2

Объем памяти ПЛИС семейства CycloneV

Table 2

Memory capacity of the CycloneV FPGA family

Обозначение семейства ПЛИС	Код	Объем памяти, Кбит		
		Блоки M10K	Блоки MLAB	Итого
Cyclone V E	A2	1760	196	1956
	A4	3080	303	3383
	A5	4460	424	4884
	A7	6860	836	7696
	A9	12200	1717	13917

Ограничение аппаратных ресурсов ПЛИС обуславливает особенности реализации нейросети для конкретной задачи. Во-первых, использование аппаратных умножителей накладывает ограничение на разрядность весовых коэффициентов нейронной сети. Это необходимо учитывать при квантовании коэффициентов в процессе проектирования и обучения сети. Во-вторых, ограничение количества аппаратных умножителей и объема встроенной памяти ПЛИС определяет вид аппаратной реализации нейросети. При небольшом количестве слоев и нейронов может быть реализована полностью параллельная архитектура сети. В иных случаях возможно итерационное послойное вычисление с повторным использованием вычислительных ресурсов на каждой итерации. Количество итераций будет зависеть от количества слоев нейронной сети. Недостаток встроенной памяти может быть компенсирован подключением внешних модулей памяти к ПЛИС. Предлагаемая структурная схема проведения вычислений в ПЛИС приведена на рис. 1.

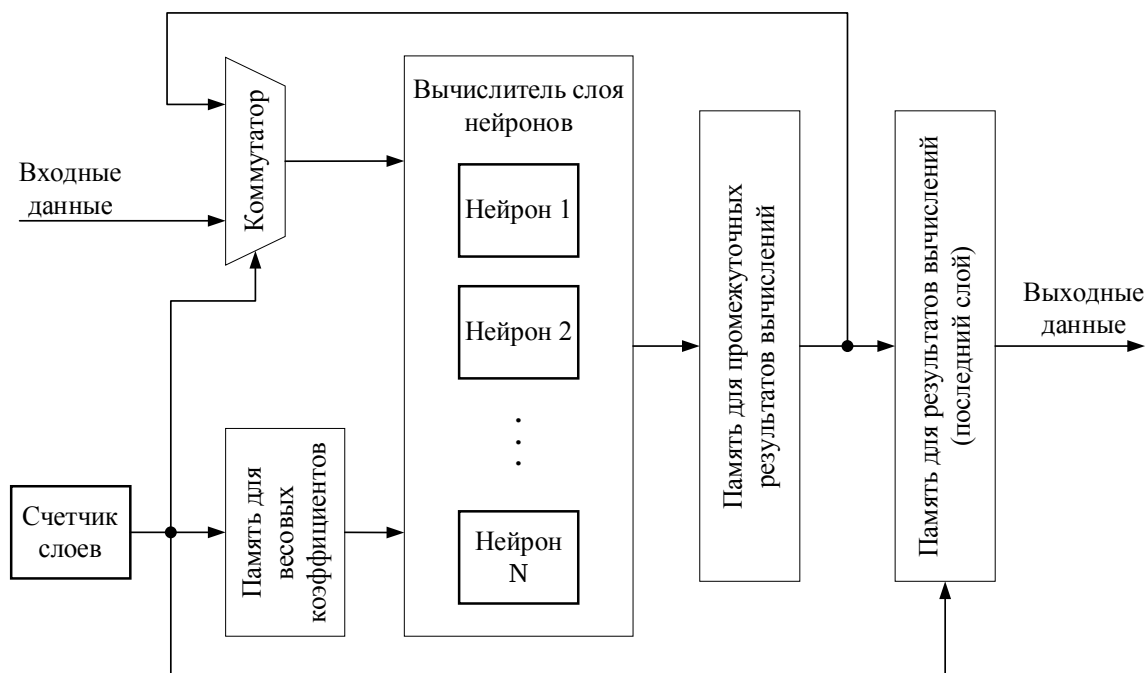


Рис. 1. Структурная схема проведения вычислений в ПЛИС
Fig. 1. Block diagram of computing in FPGA

3. Сравнение результатов обучения нейронной сети

Для обучения нейронной сети необходим набор данных для обучения (датасет). Набор данных предлагается получить путем генерации синтетических изображений, содержащих различные геометрические фигуры. Для этого был определен базовый набор фигур, включающий пять классов: стрелка, круг, квадрат, прямоугольник и треугольник (по 3000 изображений в каждом классе).

Сформированный объект накладывался на предварительно подготовленное фоновое изображение (размером 40×40 пикселей). Для каждой фигуры случайным образом определялись мас-

штаб, угол поворота и цвет. Позиционирование фигур на изображении выполнялось с отступом от краев изображения для сохранения их целостности. Некоторые объекты (15 % от общего количества) подвергались размытию с помощью фильтра Гаусса (степень размытия определялась случайным образом).

Для каждой фигуры были созданы два изображения – исходное изображение с объектом и контурная маска, представляющая собой идеальные границы объекта.

Для решения задачи классификации изображений в среде MATLAB была разработана сверточная нейронная сеть (CNN). Всего в эксперименте участвовало пять классов. Все изображения предварительно преобразовывались в градации серого с сохранением исходных размеров, что позволило сократить вычислительные затраты.

Архитектура нейронной сети для классификации включает последовательность из трех блоков, каждый из которых состоит из сверточного слоя (Convolutional Layer), слоя нормализации (Batch Normalization), функции активации (Rectified Linear Unit – ReLU) и слоя подвыборки (Max-Pooling). Сверточные слои необходимы для выделения и извлечения конкретных локальных признаков. Первый сверточный слой выделяет простые границы, второй слой – комбинации границ (например, углы, дуги и др.), третий слой – целые объекты. На рис. 2 представлена структурная схема нейронной сети.

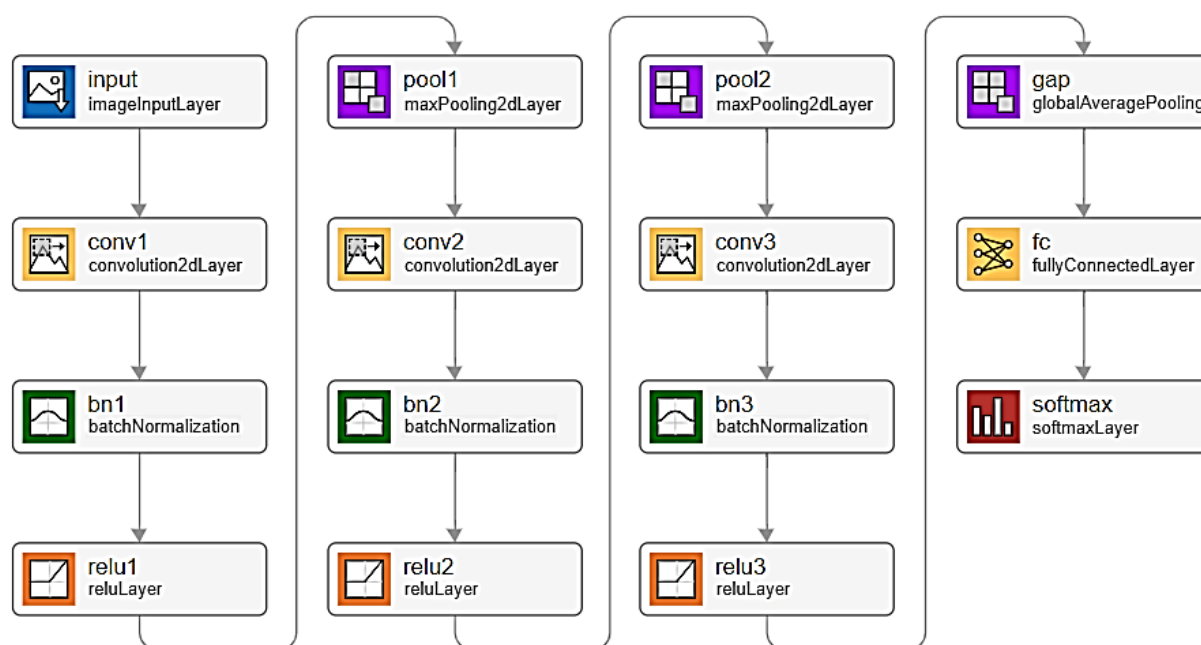


Рис. 2. Структура нейронной сети
Fig. 2. The structure of the neural network

Первый сверточный слой содержал 16 фильтров размером 3×3 , второй – 32 фильтра, а третий – 64 фильтра. Увеличение числа фильтров на глубоких слоях позволяет сети извлекать как простые локальные признаки (границы и текстуры) на начальных этапах, так и сложные формы объектов на последующих уровнях. Слой Max-Pooling позволяет уменьшить размерность входных данных для следующего слоя, что сокращает вычислительную нагрузку на последующих слоях и снижает риск переобучения. Финальный полносвязный слой содержит пять нейронов (по количеству классов) и выполняет основную работу по принятию решения к отнесению тестовых данных к какому-либо классу.

Для сравнения были обучены две нейронные сети: первая обучалась на исходных изображениях (рис. 3а), вторая – на масках изображений, содержащих идеальный контур объекта (рис. 3б). Исходные наборы данных были разделены случайным образом на три выборки: тренировочную (70 %), валидационную (15 %) и тестовую (15 %). Обучение проводилось с использованием градиентного спуска в течение 200 эпох (на одну эпоху приходится 164 итерации). Для контроля обучения каждые 100 итераций выполнялась валидация на соответствующей выборке.

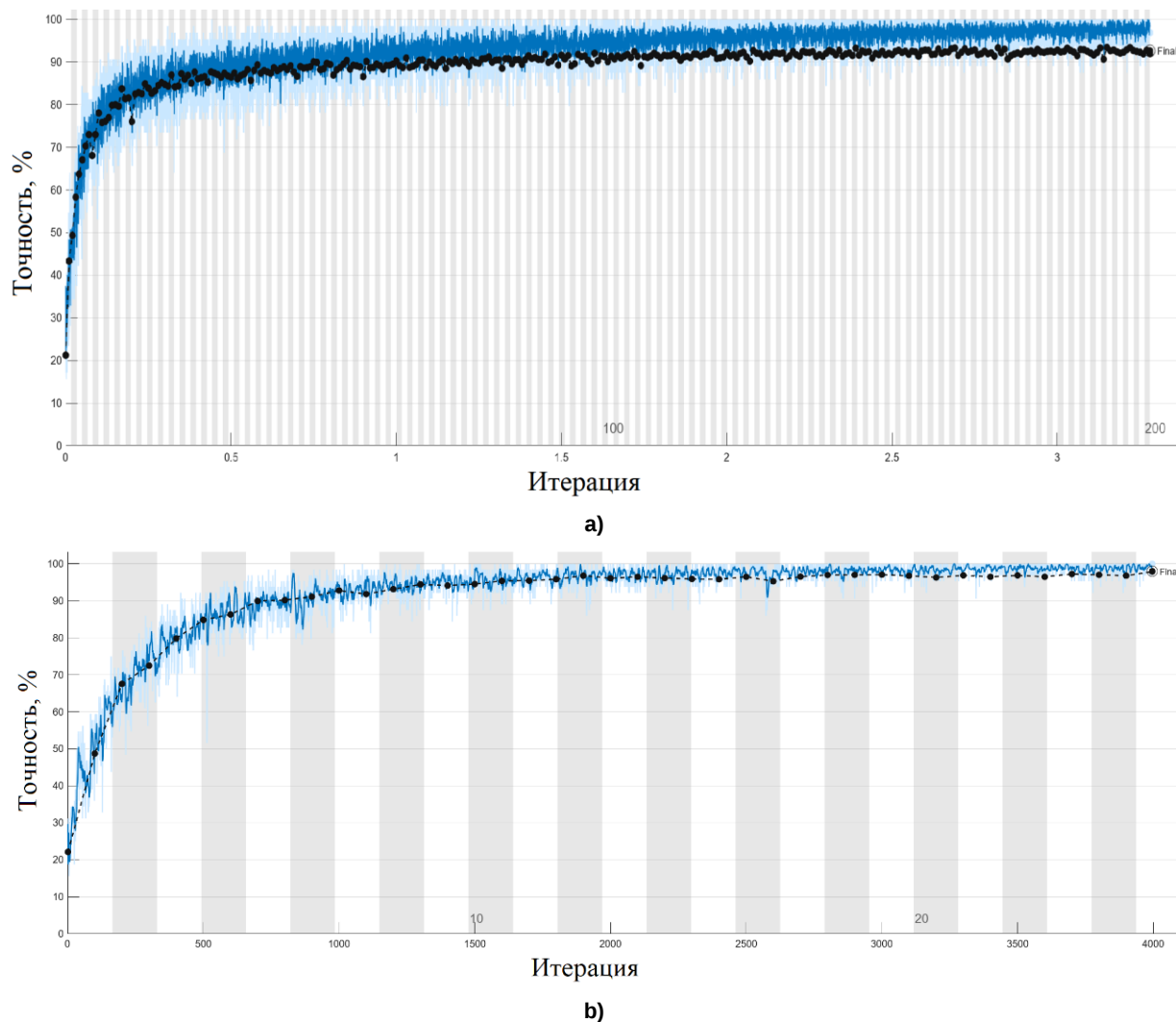


Рис. 3. Графики обучения: а – график обучения на исходных изображениях;
b – график обучения на изображениях, содержащих идеальные контуры объектов
Fig. 3. Learning schedules: a – learning schedule based on the initial images;
b – learning schedule based on images containing ideal contours of objects

После обучения нейронных сетей были получены параметры модели. В табл. 3 представлен пример весовых коэффициентов и их общее количество.

Весовые коэффициенты обученной нейронной сети

Таблица 3

Weights of a trained neural network

Table 3

Слой	Количество весовых коэффициентов	Размерность	Пример исходных весовых коэффициентов	Пример квантованных весовых коэффициентов
Convolution Layer 1	144	[3, 3, 1, 16]	[−0,3755; 0,3160; ...]	[−0,3750; 0,3125; ...]
Convolution Layer 2	4608	[3, 3, 16, 32]	[0,0762; 0,0290; ...]	[0,0781; 0,0273; ...]
Convolution Layer 3	18432	[3, 3, 32, 64]	[−0,0854; 0,0131; ...]	[−0,0859; 0,0137; ...]
Fully Connected Layer	320	[5, 64]	[0,0181; 0,2830; ...]	[0,0156; 0,2812; ...]

Тестирование обученных нейронных сетей производилось на тестовой выборке путем обработки случайных изображений (табл. 4, 5). Основное количество обработанных тестовых изображений было отнесено к истинному классу (выделены в табл. 4, 5), остальные классифицировались с ошибкой.

Таблица 4

Ошибки нейронной сети, обученной на исходных изображениях

Table 4

Errors of a neural network trained on source images

Истинный класс	Arrow	398	1	42	–	9
	Circle	1	441	2	4	2
	Rectangle	25	–	421	–	4
	Square	4	12	3	418	13
	Triangle	11	1	8	4	426
		Arrow	Circle	Rectangle	Square	Triangle
Предсказанный класс						

Таблица 5

Ошибки нейронной сети, обученной на идеальных контурах

Table 5

Errors of a neural network trained on perfect contours

Истинный класс	Arrow	438	–	9	–	3
	Circle	–	449	–	1	–
	Rectangle	9	–	441	–	–
	Square	–	17	–	443	–
	Triangle	–	1	–	9	440
		Arrow	Circle	Rectangle	Square	Triangle
Предсказанный класс						

Точность классификации сети, обученной на исходных изображениях, и сети, обученной на идеальных контурах, составила 94 и 98 % соответственно.

Нейронные сети используют память для хранения параметров (весовых коэффициентов) и активаций для каждого слоя. Большинство нейронных сетей используют типы данных с плавающей точкой и требуют значительного объема памяти и ресурсов для выполнения арифметических операций. Эти ограничения могут препятствовать применению нейронных сетей на устройствах с низкой вычислительной мощностью и малым объемом памяти. Для уменьшения объема памяти применяется квантование весовых коэффициентов в N-битные масштабированные целочисленные типы данных. Квантование представляет собой процесс преобразования параметров модели из типа данных с плавающей точкой в целочисленный формат (например, в int8). Этот процесс является ключевым для оптимизации работы нейронных сетей на устройствах с ограниченными вычислительными ресурсами. Квантование позволяет снизить требуемый для хранения параметров модели объем памяти и ускорить вычисления.

Основная идея квантования заключается в избыточности точности представления параметров. Например, весовые коэффициенты могут быть представлены с использованием меньшего количества бит без ущерба точности работы сети. Это достигается за счет масштабирования исходных значений в диапазон, который может быть представлен целочисленным типом данных, и последующего их округления.

Для каждого набора параметров вычисляется модуль максимального значения, которое используется для определения масштабирующего коэффициента. Он позволяет перенести исходные значения параметров в допустимый для целочисленного типа данных диапазон. После масштабирования значения округляются до ближайшего целого числа, что приводит к потере точности, но позволяет эффективно хранить и обрабатывать данные.

После квантования возникает задача восстановления параметров, что достигается путем умножения квантованных значений на соответствующие масштабирующие коэффициенты (см. табл. 3).

Например, квантованные весовые коэффициенты слоя «ConvolutionLayer 1» до восстановления (умножения на коэффициент 0,0078125) можно представить в виде $[-48; 40]$. Восстановленные коэффициенты являются приближенными, что может привести к снижению точности модели (рис. 4, 5).

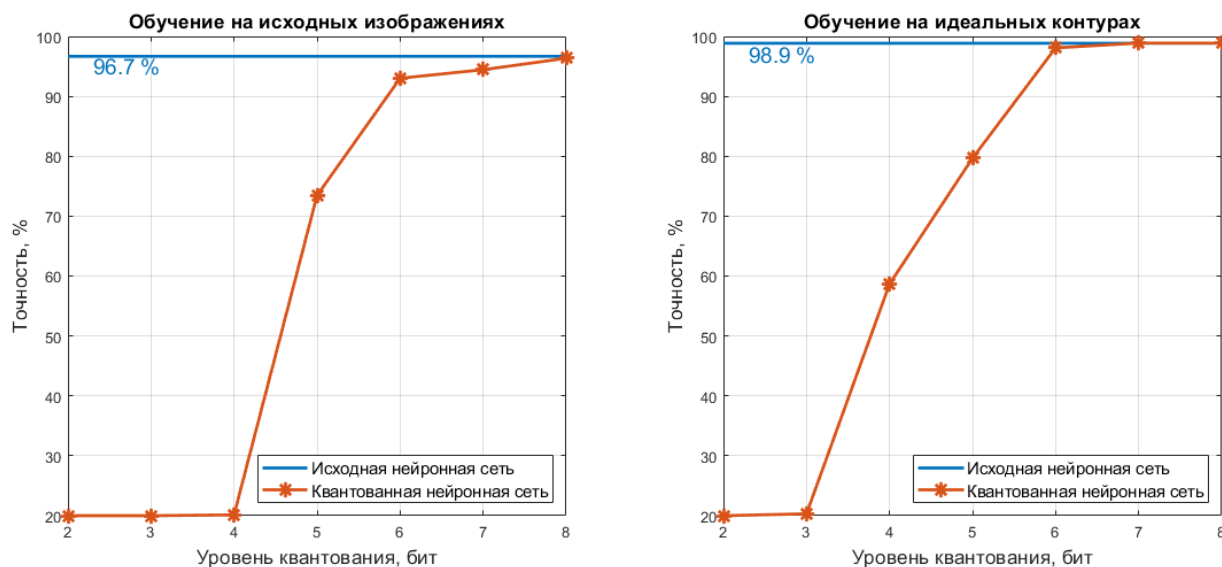


Рис. 4. Точность исходной и квантованной нейронных сетей
Fig. 4. Accuracy of the initial and quantized neural networks

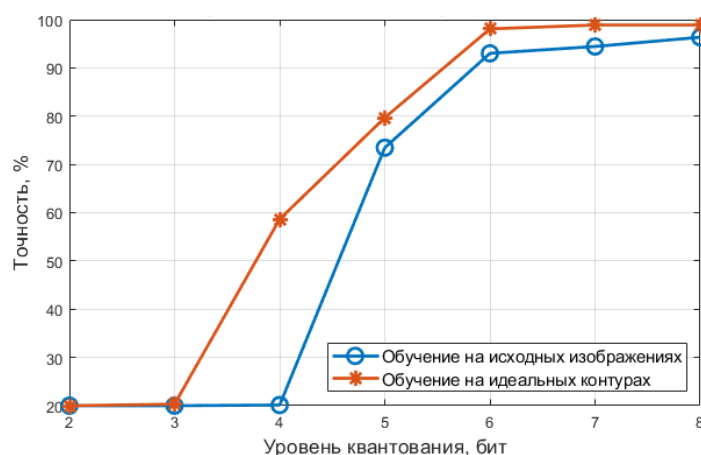


Рис. 5. Сравнение точности нейронных сетей, обученных на исходных изображениях и на контурах объектов
Fig. 5. Comparison of the accuracy of neural networks trained on the source images and on the contours of objects

Точности классификации после квантования нейронных сетей, обученных на разных типах изображений (исходных и содержащих идеальные контура объектов), отличаются. Например, при уровне квантования 4 бита нейронная сеть, обученная на контурах, показывает точность 59 %. Сеть, обученная на исходных изображениях, при этом же уровне квантования показывает точность 20 %, что соответствует случайному распределению между классами. Это подтверждает необходимость предварительной обработки данных [8, 15].

Заключение

Двухэтапный подход, сочетающий традиционные методы первичной обработки (выделение границ объектов) и нейросетевые алгоритмы вторичной обработки, позволяет найти компромисс между производительностью и вычислительной эффективностью. Это особенно важно для автономных систем с ограниченными ресурсами, которые работают в режиме реального времени. Про-

веденные исследования показали, что предварительная обработка данных существенно снижает время на обучение нейронной сети. Благодаря первичной обработке также становится возможным более грубое квантование весовых коэффициентов обученной сети. При 4-битном квантовании нейронная сеть, обученная на идеальных контурах объектов, достигает точности классификации 59 %. При этом сеть, обученная на исходных изображениях, достигает точности 20 %. Квантование весовых коэффициентов нейронной сети позволяет снизить требования к объему памяти, но в условиях ограниченных ресурсов необходимо проводить оптимизацию архитектуры нейронной сети.

Список литературы

1. Методы компьютерной обработки изображений / М.В. Гашников, Н.И. Глумов, Н.Ю. Ильясова и др.; под. ред. В.А. Сойфера. Изд. 2-е, испр. М.: Физматлит, 2003. 784 с. ISBN 5-9221-0270-2.
2. Ханова А.А., Озерова М.И. Обзор методов выделения контуров на изображениях // Информационные технологии в науке и производстве: материалы VII Всерос. молодежной науч.-техн. конф. Омск: Омский гос. техн. ун-т, 2020. С. 89–92.
3. Гонсалес Р.С., Вудс Р.Е. Цифровая обработка изображений: пер. с англ. Изд. 3-е, испр. и доп. М.: Техносфера, 2012. 1103 с.
4. Umbaugh S.E. Computer Imaging: Digital Image Analysis and Processing. CRC Press, 2005. 688 p.
5. Pratt W.K. Digital image processing. University of North Carolina, New York, USA: NY Wiley, 1978.
6. Kirsch R. Computer determination of the constituent structure of biological images // Computers and Biomedical Research. 1971. Vol. 4. P. 315–328. DOI: 10.1016/0010-4809(71)90034-6
7. Canny J. A Computational Approach for Edge Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. Vol. PAMI-8, no. 6. P. 679–698. DOI: 10.1109/TPAMI.1986.4767851
8. Особенности применения алгоритма определения границ объектов на изображениях при использовании градиентных методов / А.В. Портнов, С.Н. Даровских, А.Н. Николаев, А.Р. Николаева // Вестник УрФО. Безопасность в информационной сфере. 2024. Т. 4, № 54. С. 5–14. DOI: 10.14529/secur240401
9. TF-Net: Deploying Sub-Byte Deep Neural Networks on Microcontrollers / J. Yu, A. Lukefahr, R. Das, S. Mahlke // ACM Transactions on Embedded Computing Systems. 2019. Vol.18, iss. 5s. P. 1–21. DOI: 10.1145/3358189
10. Quantization and Deployment of Deep Neural Networks on Microcontrollers / P.-E. Novac, G. Boukli Hacene, A. Pegatoquet et al. // Sensors. 2021. Vol. 21 (9). P. 2984. DOI: 10.3390/s21092984
11. Warden P., Situnayake D. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media, Inc., 1005 Gravenstein Highway North, 2019. 484 p.
12. Omondi A.R., Rajapakse J.C. FPGA Implementations of Neural Networks. Springer, The Netherlands, 2006. 372 p.
13. Хайкин С. Нейронные сети: полный курс: пер. с англ. 2-е изд. М.: Вильямс, 2006. 1104 с. ISBN 5-8459-0890-6.
14. Cyclone V Device Handbook. Volume 1: Device Interfaces and Integration. URL: <https://www.allelcoelec.ru/datasheets.fb/5CGXFC7D6F27I7N.pdf>.
15. Портнов А.В., Николаев А.Н., Николаева А.Р. Устройство и алгоритмы первичной обработки видеoinформации для системы машинного зрения на основе искусственных фасеточных глаз // Вестник УрФО. Безопасность в информационной сфере. 2023. Т. 4, № 50. С. 5–12. DOI: 10.14529/secur230401

References

1. Gashnikov M.V., Glumov N.I., Ilyasova N.Y. et al. *Metody komp'yuternoy obrabotki izobrazheniy* [Methods of computer image processing]. 2nd ed., revised. Moscow: Fizmatlit, 2003. 784 p. (In Russ.) ISBN 5-9221-0270-2.
2. Khanova A.A., Ozerova M.I. [Overview of methods for selecting contours in images]. In: *Information technologies in science and production: Proceedings of the VII All-Russian youth scientific and technical conference*. Omsk: Omsk State Technical University, 2020. P. 89–92. (In Russ.)
3. Gonzalez R.C., Woods R.E. *Digital Image Processing*. Transl. from Engl. 3rd ed., corrected and supplemented. Moscow: Tekhnosfera, 2012. 1103 p. (In Russ.)
4. Umbaugh S.E. *Computer Imaging: Digital Image Analysis and Processing*. CRC Press, 2005. 688 p.

5. Pratt W.K. *Digital image processing*. University of North Carolina, New York, USA: NY Wiley, 1978.
6. Kirsch R. Computer determination of the constituent structure of biological images. *Computers and Biomedical Research*. 1971;4:315–328. DOI: 10.1016/0010-4809(71)90034-6
7. Canny J. A Computational Approach for Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986;PAMI-8(6):679–698. DOI: 10.1109/TPAMI.1986.4767851
8. Portnov A.V., Darovskikh S.N., Nikolaev A.N., Nikolaeva A.R. Features of the application of the algorithm for determining the boundaries of objects in images using gradient methods. *Journal of the Ural Federal District. Information security*. 2024;4(54):5–14. (In Russ.) DOI: 10.14529/secur240401
9. Yu J., Lukefahr A., Das R., Mahlke S. TF-Net: Deploying Sub-Byte Deep Neural Networks on Microcontrollers. *ACM Transactions on Embedded Computing Systems*. 2019;18(5s):1–21. DOI: 10.1145/3358189
10. Novac P.-E., Hacene G.B., Pegatoquet A., Miramond B., Gripon V. Quantization and Deployment of Deep Neural Networks on Microcontrollers. *Sensors*. 2021;21(9):2984. DOI: 10.3390/s21092984
11. Warden P., Situnayake D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, 2019. 484 p.
12. Omondi A.R., Rajapakse J.C. *FPGA Implementations of Neural Networks*. Springer, The Netherlands, 2006. 372 p.
13. Haykin S. *Neural Networks: a complete course*. Transl. from Engl. 2nd ed. Moscow: Williams, 2006. 1104 p. (In Russ.) ISBN 5-8459-0890-6.
14. Cyclone V Device Handbook. Volume 1: Device Interfaces and Integration. Available at: <https://www.allelcoelec.ru/datasheets.fb/5CGXFC7D6F2-7I7N.pdf>.
15. Portnov A.V., Nikolaev A.N., Nikolaeva A.R. Device and algorithms for primary processing of video information for a machine vision system based on artificial faceted eyes. *Journal of the Ural Federal District. Information security*. 2023;4(50):5–12. (In Russ.) DOI: 10.14529/secur230401

Информация об авторах

Вахитов Максим Григорьевич, канд. техн. наук, начальник отдела непрофильных разработок, АО «Челябинский радиозавод «Полет», Челябинск, Россия; доц. кафедры радиоэлектроники и систем связи, Южно-Уральский государственный университет, Челябинск, Россия; vakhitovmg@susu.ru.

Портнов Андрей Владимирович, аспирант кафедры радиоэлектроники и систем связи, инженер лаборатории «Цифровые системы», Южно-Уральский государственный университет, Челябинск, Россия; portnovav@susu.ru.

Николаев Андрей Николаевич, доц. кафедры радиоэлектроники и систем связи, начальник лаборатории «Цифровые системы», Южно-Уральский государственный университет, Челябинск, Россия; nikolaevan@susu.ru.

Information about the authors

Maxim G. Vakhitov, Cand. Sci. (Eng.), Head of Non-Core Development Department, JSC “Chelyabinsk Radio Plant “Polyot”, Chelyabinsk, Russia; Ass. Prof. of the Department of Radio Electronics and Communication Systems, South Ural State University, Chelyabinsk, Russia; vakhitovmg@susu.ru.

Andrey V. Portnov, Postgraduate student of the Department of Radio Electronics and Communication Systems, Engineer of the Digital Systems Laboratory, South Ural State University, Chelyabinsk, Russia; portnovav@susu.ru.

Andrey N. Nikolaev, Ass. Prof. of the Department of Radio Electronics and Communication Systems, Head of the Digital Systems Laboratory, South Ural State University, Chelyabinsk, Russia; nikolaevan@susu.ru.

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

Contribution of the authors: the authors contributed equally to this article.

The authors declare no conflicts of interests.

Статья поступила в редакцию 11.03.2025

The article was submitted 11.03.2025