DOI: 10.15514/ISPRAS-2022-34(6)-6

Математические и программные модели задач технического зрения робототехнических комплексов на основе микропроцессоров “Эльбрус”

[1] *Н.А. Бочаров, ORCID: 0000-0002-8504-2060 <bocharov.na@phystech.edu>*
[1] *Н.Б. Парамонов, ORCID: 0000-0002-6999-0968 <paramonov_n_b@mail.ru>*
[2] *О.А. Славин, ORCID: 0000-0002-9541-469X <oslavin@isa.ru>*
[1] *К.А. Суминов, ORCID: 0000-0002-3759-6026 <suminov_k@mcst.ru>*

[1] *Институт электронных управляющих машин им. И.С. Брука, 119334, Россия, г. Москва, ул. Вавилова, д. 24*
[2] *Федеральный исследовательский центр "Информатика и управление" РАН, 117312, Москва, проспект 60-летия Октября, 9*

Аннотация. Создание новых поколений автономных робототехнических комплексов, систем распознавания и систем технического зрения в целом невозможно без использования современных компьютерных технологий. В данной статье представлены модели системы технического зрения роботов на базе микропроцессоров "Эльбрус". Были разработаны модели задач обнаружения, классификации и сегментации. Теоретические и экспериментальные результаты были получены на существующих и перспективных микропроцессорах "Эльбрус". Показано, что микропроцессоры "Эльбрус" могут быть основой бортовой системы технического зрения. Полученные авторами результаты свидетельствуют о перспективах импортозамещения в области робототехники.

Ключевые слова: моделирование; Эльбрус; e2k; VLIW; бортовой вычислитель; техническое зрение; робототехника

Для цитирования: Бочаров Н.А., Парамонов Н.Б., Славин О.А., Суминов К.А. Математические и программные модели задач технического зрения робототехнических комплексов на основе микропроцессоров “Эльбрус”. Труды ИСП РАН, том 34, вып. 6, 2022 г., стр. 85-100. DOI: 10.15514/ISPRAS-2022-34(6)-6

Mathematical and software models of technical vision tasks of robotic complexes based on “Elbrus” microprocessors

[1] *N.A. Bocharov, ORCID: 0000-0002-8504-2060 <bocharov.na@phystech.edu>*
[1] *N.B. Paramonov, ORCID: 0000-0002-6999-0968 <paramonov_n_b@mail.ru>*
[2] *O.A. Slavin, ORCID: 0000-0002-9541-469X <oslavin@isa.ru>*
[1] *K.A. Suminov, ORCID: 0000-0002-3759-6026 <suminov_k @mcst.ru>*

[1] *PJSC “INEUM named after I.S. Brook”, 24, Vavilova st., Moscow, 119334, Russia*
[2] *ISA FRC RAS, 9, 60-letiya Oktyabrya av., Moscow, 117312, Russia*

Abstract. The creation of new generations of autonomous robotic complexes, recognition systems and vision systems in general is impossible without the use of modern computer technologies. This article presents models

of the robot vision system based on Elbrus microprocessors. Models of detection, classification and segmentation tasks were developed. The models are based on the number of arithmetic operations required to perform a forward pass. The models take into account such features of Elbrus microprocessors as: number of executing devices, pipeline, data pre-pumping, clock frequency, etc. Theoretical and experimental results were obtained on existing and promising "Elbrus" microprocessors. It is shown that Elbrus microprocessors can be the basis of an on-board vision system. The results obtained by the authors indicate the prospects of import substitution in the field of robotics.

Keywords: modeling; Elbrus; e2k; VLIW; onboard computer; technical vision; robotics

For citation: Bocharov N.A., Paramonov N.B., Slavin O.A., Suminov K.A. Mathematical and software models of technical vision tasks of robotic complexes based on “Elbrus” microprocessors. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 6, 2022. pp. 85-100 (in Russian). DOI: 10.15514/ISPRAS-2022-34(6)-6

1. Введение

Задачи технического зрения в настоящее время являются важным направлением развития области искусственного интеллекта [1]. Создание новых поколений автономных робототехнических комплексов (РТК), систем распознавания и систем технического зрения в целом невозможно без использования современной вычислительной техники. При этом для решения подобного рода задач в настоящее время активно применяются и разрабатываются [2-4] вычислительные комплексы с использованием специализированных ускорителей. Использование таких ускорителей обусловлено неспособностью микропроцессоров (МП) общего назначения решить такие задачи за поставленное время вследствие большой вычислительной нагрузки.

Тем не менее, граница применимости таких специализированных ускорителей при проектировании вычислительных комплексов для решения подобных задач часто определяется эмпирически, в особенности для вычислительных комплексов (ВК) на основе МП серии «Эльбрус» [5], поскольку, в силу особенности архитектуры МП «Эльбрус» (Very Long Instruction Word, VLIW), сложно и не всегда возможно оценить сложность и ресурсоемкость решения на базе имеющихся решений подобных задач, реализованных для систем с МП других архитектур.

Одним из важных и актуальных применений бортовых систем с МП серии «Эльбрус» являются бортовые вычислители и системы технического зрения. В ходе проведенных в МЦСТ работ показано, что вычислители на основе МП серии «Эльбрус» могут и успешно используются для решения задач технического зрения как в серверном [6] так и в бортовом [7] режимах. В 2017 году в МЦСТ разработан и внедрен бортовой вычислитель на базе МП Эльбрус-4С, обеспечивающий достаточную производительность для автономного движения робота на скоростях до 40 км\ч. В работе [8] было обосновано, что для обеспечения корректной работы на такой скорости достаточным условием является работа системы технического зрения с производительностью не менее 10 кадров в секунду.

Появление новых МП серии «Эльбрус», таких как Эльбрус-8СВ, Эльбрус-2С3 и Эльбрус-16С [9], а также средств вычислительной техники на их основе [10] открывает новые перспективы перед разработчиками РТК. Высокая производительность, обеспечиваемая новыми МП серии «Эльбрус», позволит создавать бортовые вычислительные комплексы, способные решать задачи технического зрения на РТК с использованием отечественной программно-аппаратной платформы, а появление МП «Эльбрус» шестого поколения должно еще больше повысить производительность существующих решений и открыть возможности для решения новых задач в этой области.

Одними из самых частых задач в области технического зрения являются задачи сегментации, обнаружения и классификации объектов на изображении. Для обнаружения объектов одним из популярных методов, находящих широкое применение, является метод Виолы-Джонса (Viola–Jones object detection) [11]. Для решения задачи классификации, как и задач

сегментации изображений, как правило, с большим успехом используются сверточные нейронные сети различных архитектур.

Специальные ускорители активно создаются с целью ускорения решения, в том числе, именно таких задач, но в силу дороговизны, большей сложности и ограничений в использовании систем со специализированными ускорителями встает вопрос об определении условий, в которых для решения рассматриваемых задач достаточно использовать МП общего назначения, например, из ряда «Эльбрус», без специальных ускорителей.

2. Особенности аппаратно-программной платформы «Эльбрус» для решения задач технического зрения

Архитектура процессоров VLIW является альтернативой для архитектур OOOSS (Out-Of-Order SuperScalar), главным отличиями которой является использование так называемых широких команд, позволяющих выразить параллельность множества операций в ассемблере, а также использование оптимизирующих компиляторов [12], переупорядочивающих команды во время компиляции программы, а не во время ее выполнения.

Компилятор для VLIW обладает гораздо большим окном операций для перемешивания, чем имеется на этапе исполнения у аппаратуры OOOSS. Это позволяет в некоторых случаях лучше выявлять независимые операции для их параллельного исполнения. С другой стороны, OOOSS обладает дополнительной информацией о параллелизме, доступной в динамике исполнения, например, значения адресов операций чтения и записи. Это позволяет лучше выявлять параллелизм в некоторых других ситуациях.

Широкие команды в МП серии «Эльбрус» содержат набор элементарных операций, которые можно запустить на исполнение в одном процессорном такте. Для широких команд процессоров «Эльбрус» с системой команд версии не ниже 4 доступны 6 арифметико-логических устройств (АЛУ), поддерживающих операции с вещественными числами, устройство передачи управления, 3 устройства для работы с предикатами, 6 квалифицирующих предикатов, 4 устройства для команд асинхронного чтения данных – APB (Array Prefetch Buffer), 4 32х-битных литерала для хранения константных значений – LIT. Состав широкой команды представлен на рис. 1 [13].

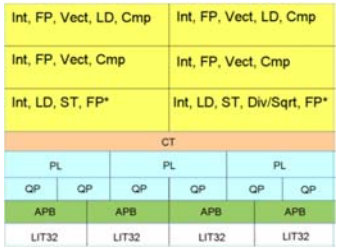

Рис. 1. Парк устройств широкой команды МП «Эльбрус»
Fig. 1. The list of devices of the wide command of MP "Elbrus"

Ядро поддерживает большое количество операций – 25 в скалярном и 41 в векторном режимах. Также стоит отметить, что АЛУ в составе ядра поддерживают выполнение зацепленных друг за другом операций в качестве одной трехаргументной операции, например, операции вида a * b + c. При этом промежуточный результат первой операции не записывается в регистр, а сразу используется в качестве аргумента второй операции. Наличие такой операции является важным при работе со сверточными нейронными сетями, в которых такая комбинация операций является основополагающей для работы всего алгоритма.

Архитектура «Эльбрус» включает в себя многие решения, которые обеспечивают улучшение производительности при работе со сверточными нейронными сетями, в том числе:

- конвейеризация циклов позволяет наиболее эффективно исполнять циклы с независимыми (или слабо зависимыми) итерациями; в программно-конвейеризированном цикле последовательные итерации выполняются с наложением - одна или несколько следующих итераций начинают выполняться раньше, чем заканчивается текущая;
- методы предварительной подкачки данных – в том числе устройство асинхронной подкачки массивов APB; устройство применяется для асинхронной предподкачки необходимых элементов массива заранее перед использованием, что ускоряет процесс обращения в память во время исполнения программы и уменьшает время ее работы делая более эффективной работу при большом количестве итераций; APB особенно актуально при работе со сверточными нейронными сетями, например, для заблаговременной подкачки очередной порции весовых коэффициентов (ядер сверток).

Также для оптимизации выполнения команд применяются такие техники, как unroll&fuse (unroll&jam) [14], применяемые компилятором для попытки развернуть и слить выполнение цикла там, где это возможно. При этом развертывание цикла повторяет тело нескольких циклов и объединяет необходимые итерации в один развернутый цикл. Использование объединенного цикла может свести к минимуму необходимое количество итераций и снизить частоту промахов в кэш, что актуально при больших объемах параметров-весов нейронных сетей.

Кроме архитектурных решений и механизмов, для МП архитектуры «Эльбрус» реализованы и портированы программные решения и библиотеки, в их числе библиотека EML – Elbrus Media Library, высокопроизводительная математическая и мультимедийная библиотека, представляющая из себя набор разнообразных функций для обработки сигналов, изображений, видео, математических вычислений. Благодаря эффективной реализации библиотеки EML, на некоторых задачах с матрицами с ее использованием ускорение составляет более 20 раз по сравнению с процессором Intel и до 85 раз по сравнению с процессором «Эльбрус» без использования библиотеки EML [15].

Библиотека EML содержит в своем составе разделы:

- Core – для работы с памятью;
- Vector – для работы с векторами;
- Algebra – раздел линейной алгебры включающий пакеты для работы с матрицами и векторами BLAS 1, BLAS 2, BLAS 3, LAPACK.

Это позволяет эффективно выполнять операции линейной алгебры, в том числе с матрицами, что необходимо при работе со сверточными нейронными сетями.

Кроме библиотеки EML, для МП архитектуры «Эльбрус» портирована библиотека Opencv 3.2.0. В настоящее время находится на завершающем этапе работа по портированию для архитектуры «Эльбрус» актуальной версии библиотеки OpenCV 4.5.2, которая также включает в себя модули Opencv_dnn для работы с глубокими нейронными сетями.

На завершающем этапе портирования находится фреймворк для работы с машинным обучением PyTorch для языка Python. Фреймворк PyTorch является одним из самых популярных в мире и активно применяется, в том числе, для решения задач с применением сверточных нейронных сетей различных архитектур. Благодаря его наличию, для МП архитектуры «Эльбрус» становится возможным использование существующих решений задач, а также создание универсальных решений для различных архитектур на его базе.

Кроме того, для архитектуры «Эльбрус» закончена стадия практического тестирования отечественного фреймфорка для обучения и работы с глубокими нейронными сетями – платформа ГНС [16]. Платформа представляет собой клиент-серверное приложение, позволяющее пользователям работать с эффективными реализациями нейронных сетей различных архитектур. Совместно с платформой разработчиком поставляется библиотека, учитывающая архитектурные особенности процессоров семейства «Эльбрус».

Таким образом, большой парк вычислительных устройств (АЛУ) в составе процессоров «Эльбрус», большое количество операций – 25 в скалярном и 41 и векторном режимах [17] за такт на одно ядро (в Эльбрус-8С), такие механизмы, как предварительная подкачка данных, возможность выполнять совмещенные операции и эффективная работа с матрицами, а также большое количество ядер МП дают возможность поставить процессоры «Эльбрус» в ряд между процессорами общего назначения и специализированными процессорами, использующими SIMD-инструкции. А наличие оптимизированных библиотек и фреймворков, обеспечивающих эффективную работу с матрицами и другими математическими вычислениями в совокупности с большой степенью параллельности многих задач, обеспечивающих функционирование систем технического зрения, дают основание полагать, что задачи подобного класса могут эффективно решаться и использованием современных МП серии «Эльбрус».

3. Модель задачи обнаружения

Алгоритмы выделения и распознавания образов на изображении, работающие на основе дерева классификаторов, способны эффективно обнаружить на изображении объекты, совпадающие по своим свойствам, например, форме и цвету, с заранее заданными программистом объектами. Таким образом, эти алгоритмы могут использоваться для поиска заранее известных объектов на изображении. Одним из наиболее распространенных подобных алгоритмов является алгоритм Виолы-Джонса.

Основными операциями при работе алгоритма являются операции умножения и сложения. Произвести оценку количества выполняемых операций сложения и умножения S и M соответственно, можно согласно формуле:

$$S = M = \frac{A * B}{w * h * s^2} * \left(\frac{w * h}{w_0 * h_0}\right)^2 * \sum_{i=0}^{N} S_i * f_i,$$

где A – ширина входного изображения, A – высота входного изображения, w – ширина поискового окна, h – высота поискового окна, w_0 – ширина базового поискового окна, h_0 – высота базового поискового окна, s – относительный размер шага поискового окна, S_i – результат *i*-го классификатора, представленный значениями 0 и 1 при отсутствии и наличии объекта соответственно, f_i – количество признаков на i-м этапе классификатора N – количество классификаторов в каскаде.

При оценке самого вычислительно нагруженного случая, когда все классификаторы на изображении обнаруживают объект и проводятся все вычисления, количество рассчитываемых признаков составляет более 6000 для каждого из них. При оценке самого вычислительно ненагруженного случая, когда все классификаторы отбрасывают окно на первом этапе, и останавливают дальнейшие вычисления, количество рассчитанных признаков составляет 10 для каждого из них.

Для изображения размером 1280x720 пикселей теоретическая оценка производительности составляет от 1,7 кадра в секунду в наиболее нагруженном и до 1100 кадров в наименее нагруженном случаях. В действительности, за счет каскадной архитектуры классификатора количество выполняемых операций значительно варьируется в зависимости от содержания изображения. Вследствие очень широкого диапазона теоретической оценки производительности для более точной оценки требуются экспериментальные исследования на различных видеорядах.

Для моделирования задачи обнаружения были разработаны программы обнаружения объектов в видеопотоке с использованием метода Виолы-Джонса. Разработаны программы для использования на процессорах архитектуры «Эльбрус», на процессорах Intel, а также для использования на процессорах семейства Intel совместно с видеокартой Nvidia GTX 960 в качестве ускорителя. Программы написаны с использованием языка C++, библиотеки

OpenCV версии 4.5.2 для архитектуры Intel x86-64 и архитектуры «Эльбрус». Также использовались некоторые идеи, опубликованные в [18, 19]. Для версии, использующей видеокарту Nvidia в качестве ускорителя, также использовался фреймворк Nvidia CUDA toolkit версии 10.0 [20].

В библиотеке OpenCV применялся модуль objdetect в версиях, ориентированных на исполнение на универсальных процессорах Intel и «Эльбрус», а также модуль cudaobjdetect содержащий реализации алгоритмов и функций на основе модели программирования CUDA, использующей в качестве вычислителя видеокарту. В указанных модулях для обнаружения объектов использовался класс CascadeClassifier, в том числе, метод этого класса detectMultiScale, применяющий алгоритм Виолы-Джонса к изображению и выполняющий обнаружение объектов разных размеров на изображении. Результатом работы метода является список прямоугольников, заданных координатами на исходном изображении и ограничивающих обнаруженные объекты, если таковые нашлись.

При разработке программной модели для оценки эффективности решения были проведены тесты для различных размеров скользящих окон с различным соотношением сторон, в том числе, квадратных. В результате для дальнейших экспериментов были выбраны окна различных размеров с соотношением сторон 23:28, как дающие наилучший результат. Также были протестированы каскадные классификаторы с различными наборами признаков, в результате чего были выбраны классификаторы haarcascade_frontalface_alt и cuda_haarcascade_frontalface_alt для реализации на универсальном процессоре и с использованием CUDA соответственно.

4. Модель задачи классификации

Среди архитектур сверточных нейронных сетей, дающих хороший результат top 5 accuracy (более 92%) на соревновании ImageNet [21], являются сети VGG16 и VGG19 [22]. Кроме того, сверточная часть этих сетей, в особенности VGG16, часто используется для предварительной обработки (feature detector) для выделения признаков, которые в дальнейшем используются в других приложениях.

Для теоретического обоснования времени выполнения вычислений нейронной сети с архитектурой VGG16 и VGG19, на примере МП серии «Эльбрус», разработана математическая модель вычислений, учитывающая количество операций, производимых при расчетах сети. В табл. 1 представлено необходимое количество параметров и их объем, а также количество и тип необходимых операций при выполнении основных расчетов для VGG19 и VGG16. Слои, помеченные (*), не входят в VGG16.

Табл. 1. Количество параметров и операций для слоев сети VGG19 (VGG16)
Table 1. Number of parameters and operations for network layers VGG19 (VGG16)

№ слоя	слой	размерность входа			количество умножений в слое	количество сложений в слое	количество сравнений в слое	количество параметров в слое
		ш	в	г				
1	conv3-64	224	224	3	89915392	86704128	0	1792
2	conv3-64	224	224	64	1852899328	1849688064	0	36928
	pool2	224	224	64	0	0	2408448	0
3	conv3-128	112	112	64	926449664	924844032	0	73856
4	conv3-128	112	112	128	1851293696	1849688064	0	147584
	pool2	112	112	128	0	0	1204224	65664
5	conv3-256	56	56	128	925646848	924844032	0	295168
6	conv3-256	56	56	256	1850490880	1849688064	0	590080
7	conv3-256	56	56	256	1850490880	1849688064	0	590080
8(*)	conv3-256	56	56	256	1850490880	1849688064	0	590080
	pool2	56	56	256	0	0	602112	0
9	conv3-512	28	28	256	925245440	924844032	0	1180160

10	conv3-512	28	28	512	1850089472	1849688064	0	2359808
11	conv3-512	28	28	512	1850089472	1849688064	0	2359808
12(*)	conv3-512	28	28	512	1850089472	1849688064	0	2359808
	pool2	28	28	512	0	0	301056	0
13	conv3-512	14	14	512	462522368	462422016	0	2359808
14	conv3-512	14	14	512	462522368	462422016	0	2359808
15	conv3-512	14	14	512	462522368	462422016	0	2359808
16(*)	conv3-512	14	14	512	462522368	462422016	0	2359808
	pool2	14	14	512	0	0	75264	0
17	fc4096	1	1	25088	102760448	102760448	0	102764544
18	fc4096	1	1	4096	16777216	16777216	0	16781312
19	fc1000	1	1	4096	4096000	4096000	0	4097000
		Всего VGG19			19646914560	19632062464	4290048	143732904
		Всего VGG16			15483811840	15470264320	4591104	138423208

В табл. 2 представлены значения теоретической оценки времени выполнения расчетов сетей (inference) при рассмотрении идеальной модели процессоров, в которой отсутствуют задержки по памяти, длительность вычислений ограничивается лишь скоростью работы и степенью конвейеризации АЛУ. В действительности, при наличии не идеальности процессов вычисления и существующих потерь, а также ограниченного размера кэш-памяти использование механизма предподкачки, реализованного в МП серии «Эльбрус» [23] механизма APB, а также эффективная реализация умножения матриц блоками, использование схемы unroll and fuse и возможность АЛУ выполнять зацепленные операции умножения и сложения позволяют получить скорость предподкачки, обеспечивающую на ~90% эффективную загруженность АЛУ. Время выполнения T в этом случае можно оценить, как:

$$T = \frac{(Nu + Nm + Ns + Nc) * R * k}{F * C * A * S},$$

где Nu – количество совмещенных операций, Nm, Ns и Nc – количество отдельных операций умножения, сложения и сравнения соответственно, R – разрядность чисел, k – коэффициент эффективности, F – тактовая частота процессора, C – количество ядер, A – количество АЛУ включающих FPU (Floating Point Unit), S – разрядность FPU.

Табл. 2. Теоретическое время выполнения вычислений для сетей VGG16 и VGG19 на МП «Эльбрус»
Table 2. Theoretical execution time for VGG16 and VGG19 networks on MP «Elbrus»

	VGG16				**VGG19**			
количество совмещенных операций * +	15470264320				19632062464			
количество операций *	13547520				14852096			
количество операций +	0				0			
количество операций <>	4290048				4591104			
процессор	Эльбрус-8C	Эльбрус-8CB	Эльбрус-16C	Эльбрус-2C3	Эльбрус-8C	Эльбрус-8CB	Эльбрус-16C	Эльбрус-2C3
расчетное время расчета прямого прохода (мс)	137,9	59,7	22,4	179,2	174,9	75,8	28,4	227,4
расчетное количество кадров в секунду	7,3	16,7	44,6	5,6	5,7	13,2	35,2	4,4

Для моделирования задачи классификации были разработаны программы, реализующие вычисления нейронных сетей различных архитектур. Для задачи классификации с использованием архитектуры нейронной сети VGG16 были разработаны следующие программы: универсальная программа на языке C без внешних и архитектурных зависимостей, программа на языке Python 3 с использованием фреймворка машинного обучения с открытым кодом PyTorch для вычислений на универсальных процессорах и видеокарты Nvidia GTX 960 в качестве ускорителя. Также разработана программа с использованием платформы ГНС разработки ГосНИИАС в которой были реализованы вычисления нейронных сетей архитектуры VGG19, а также некоторых других распространенных архитектур. При реализации программ использовались некоторые идеи, опубликованные в [17, 24].

Универсальная программа на языке C осуществляет вычисления прямого прохода (inference) нейронной сети архитектуры VGG16 для изображений размером 224 x 224 x 3. Для реализации версии программы на языке C были использованы только стандартные библиотеки из состава языка программирования C без архитектурных зависимостей, в результате чего программа пригодна для вычислений на универсальных процессорах разных архитектур, т.е. является полностью кроссплатформенной. Также были использованы файл предобученных весов нейронной сети с соревнований ImageNet и файл расшифровки классов. Классификация выполняется среди 1000 классов соревнований ImageNet. При реализации программы, ввиду отсутствия использования архитектурных зависимостей библиотеки EML, позволяющих получить ускорение, было получено большее время обработки одного изображения, чем ожидалось.

Также была разработана модель задачи классификации с использованием Платформы-ГНС. Файлы обученных нейронных сетей для архитектур VGG19, AlexNet, LeNet, ResNet18, ResNet50, MobileNetV1 были обучены и предоставлены ГосНИИАС. Для работы с обученными файлами в комплекте поставляется библиотека, использующая архитектурные зависимости.

5. Модель задачи сегментации

Предложенная Адамом Пашке (Adam Paszke) и др. в работе [25] архитектура нейронной сети с названием ENet была создана для решения задачи сегментации с малыми затратами вычислительных ресурсов. Архитектура сети ENet представляет собой архитектуру кодер-декодер, и ее работа делится на 6 этапов, строящихся из блоков нескольких типов – это initial block, asymmetric bottleneck block, upscale bottleneck block, bottleneck block в трех вариантах, а также fullconvolution block.

Согласно расчетам, для изображения размерностью 640x360x3 необходимы базовые операции в количестве, указанном в табл. 3. Количество указано в миллионах операций.

Табл. 3. Количество основных необходимых операций
Table 3. Number of basic operations required

умножения	сложения	сравнения	деления	извлечение корня
2386	2472	58	30	30

Таким образом, всего необходимо около 5 миллиардов операций. Учитывая возможность выполнять зацепленные совмещенные операции, а также временную зависимость некоторых данных, в общей сложности получается 3.8 млрд операций [25] и 370 тысяч параметров.

Аналогично расчетам для модели сетей VGG19, VGG16, в табл. 4 представлены теоретические оценки времени выполнения расчетов сети (inference) при рассмотрении

идеальной модели процессоров, а также с учетом не идеальности процессов вычисления и существующих потерь, а также размера кэш-памяти, с использованием механизма предподкачки, реализованного в МП серии «Эльбрус», механизма APB, а также при эффективной реализации умножения матриц блоками, использования схемы unroll and fuse и возможности АЛУ выполнять зацепленные операции умножения и сложения. Скорость предподкачки в таком случае достигает значения, обеспечивающего на ~90% эффективную загруженность АЛУ. Время выполнения T в этом случае можно оценить, как:

$$T = \frac{(Nu + Nm + Ns + Nc + Nd + Nq) * R * k}{F * C * A * S},$$

где Nu –количество совмещенных операций, Nm, Ns, Nc, Nd и Nq – количество отдельных операций умножения, сложения, сравнения, деления и вычисления квадратного корня соответственно, R – разрядность чисел, k – коэффициент эффективности, F – тактовая частота процессора, C – количество ядер, A – количество АЛУ, включающих FPU, S – разрядность FPU.

Также в табл. 4 представлена оценка времени для видеокарты GTX 960 и ВК Nvidia tx1 используемого авторами архитектуры ENet в своих тестах.

Табл. 4. Теоретическое время выполнения вычислений для сети ENet
Table 4. Theoretical execution time for the ENet

количество совмещенных операций * +	752					
количество отдельных операций *	441					
количество отдельных операций +	1720					
количество отдельных операций < >	58					
количество отдельных операций /	30					
количество отдельных операций √	30					
вычислитель	Эльбрус-8C	Эльбрус-8CB	Эльбрус-16C	Эльбрус-2C3	GTX 960	Nvidia TX 1
теоретическое время расчета прямого прохода сети (мс)	15,6	6,8	2,5	20,3	1,6	7,6
теоретическое количество кадров в секунду	64,0	147,7	393,8	49,2	618,7	131,3

Для моделирования задачи сегментации изображений была разработана программа сегментации изображений с использованием нейронной сети архитектуры ENet. Программа написана с использованием языка C++, библиотеки OpenCV версии 4.5.2 для архитектуры Intel x86-64 и архитектуры «Эльбрус». Также использовались некоторые идеи, опубликованные в [26-28].

В библиотеке OpenCV использовался модуль opencv_dnn для работы с глубокими нейронными сетями. Также использовались файлы весов предобученной сети Enet.

6. Общая модель технического зрения робототехнических комплексов

На основе разработанных программ для задач обнаружения, классификации и сегментации была разработана единая программная модель, объединяющая в себе все указанные модели, общую программу тестирования, скрипты сборки для Cmake и примеры входных данных. Общая модель написана на языке python и позволяет провести общее тестирование системы

на соответствие необходимым параметрам производительности. Общий объем тестирующей системы с примерами исходных данных для программ составил 1.5 Гб.

Для тестирования разработанных программ применялись различные процессоры семейства «Эльбрус», в том числе 2, 8 и 16 ядерный, а также МП Intel core i7 2600k, серверные МП Intel Xeon 4110 и Xeon e5 2620, выпуска 2016-2017 года, 8 ядерные, 16 поточные и мобильный процессор Intel core i7-8565U производства 2018 года. В качестве спец. ускорителя была использована видеокарта Nvidia GeForce GTX 960 совместно с МП Intel core i7 2600k.

Для проведения эксперимента с обнаружением был выбран один видеоряд со следующими разрешениями: 424 x 240, 640 x 360, 854 x 480, 1280 x 720, 1920 x 1080, 2560 x 1440, 3840 x 2160, также производилось тестирование, и отладка на видеопотоке с веб-камеры в реальном времени.

В качестве входных данных для всех программ, моделирующих вычисления нейронной сети с архитектурой VGG16, использовались изображения из базы данных ImageNet с разрешением 224x224x3. Для нейронных сетей с архитектурами VGG19, AlexNet, LeNet, ResNet18, ResNet50, MobileNetV1 кроме изображений с разрешением 224x224x3, также были использованы изображения в разрешении 32x32x3.

Для задачи сегментации в качестве исходных данных было выбрано видео, снятое на видеорегистратор автомобиля и содержащее движение автомобиля по дорогам общего пользования, в разрешении 1280x720 и наборы кадров, полученные из этого видео. Также производилось тестирование и отладка на видеопотоке с веб-камеры в реальном времени.

При моделировании задачи обнаружения объектов в видеопотоке были получены результаты производительности в зависимости от разрешения изображения для различных размеров поискового окна. На рис. 2 представлены графики зависимости производительности, выраженной в количестве кадров в секунду от разрешения входного видеопотока для вычислительных комплексов на базе различных процессоров, в том числе для ВК, включающего ускоритель GTX 960. Результаты представлены для поискового окна размером 92 x 112.

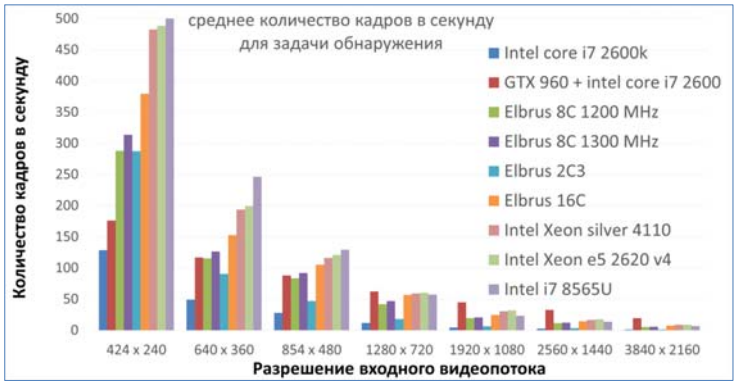

Рис 2. Количество кадров в секунду при решении задачи обнаружения
Fig 2. The number of frames per second when solving the detection problem

Из полученных данных видно, что все протестированные процессоры семейства «Эльбрус» стабильно превосходят Intel core i7 2600k, в том числе двухядерный Эльбрус-2C3. Из результатов видно, что на малых разрешениях использование совместно с процессором ускорителя в виде видеокарты не только не дает выигрыша по времени, но и обеспечивает худшие результаты в отличии от использования только процессора, что связано с особенностью распределенных вычислений с использованием спецускорителей, а именно, с длительной и частой пересылкой маленьких для расчета объемов данных между процессором

и ускорителем. Однако стоит заметить, что в ходе исследования использовалась относительно устаревшая видеокарта, и при использовании современных спецускорителей типа Модуль [30], Элвис [31], Nvidia разница в результатах будет отличаться. Также в ходе эксперимента установлено, что Эльбрус-8С превосходит вычислитель на основе Intel core i7 2600k совместно с видеокартой GTX 960 на разрешениях вплоть до 854x480, а на разрешении 1280x720 Эльбрус-8С показал результат до 47 кадров в секунду, при том, что Эльбрус-2С3 обеспечивает до 18, а Эльбрус-16С до 57 кадров в секунду, в то время как Intel core i7 2600 – всего около 11 кадров, а совместно с видеокартой – 62 кадра в секунду. Эльбрус-16С показал сравнимый с серверными процессорами Intel Xeon результат на больших разрешениях входного видеопотока, а Эльбрус-8С показал отставание от них на 10-30% при вдвое меньшем количестве потоков.

При моделировании задачи классификации изображений для всех программ моделирования расчетов нейронных сетей и используемых архитектур нейронных сетей были получены значения времени выполнения. Для программы реализации вычислений сети VGG16 на языке С были получены результаты, представленные в табл. 6.

Табл. 6. Время выполнения задачи классификации VGG16
Table 6. VGG16 classification task execution time

Процессор	**Эльбрус-2С3**	**Эльбрус-Е8С**	**Эльбрус-Е16С**	**Intel i7-2600k**	**Intel xeon silver 4110**	**Intel xeon e5 2620 v4**
однопоточный режим (мс)	34,2	56,4	31,3	16,3	15,7	15,7
многопоточный режим (мс)	18,7	9	3,6	4,4	1,9	2,5

Для реализации сети VGG16 на PyTorch для процессоров из семейства «Эльбрус» были получены результаты для МП Эльбрус-8С. Для других процессоров линейки на данный момент отсутствует реализация фреймворка PyTorch. Также был получен результат для процессора Intel i7 2600k и Intel i7 2600k совместно с ускорителем Nvidia GTX 960. Результаты представлены в табл. 7, значения выражены в миллисекундах.

Таблица 7. Время выполнения задачи классификации VGG16 на Python
Table 7. Execution time of the VGG16 classification task in Python

	Intel i7 2600	**Intel i7 2600 + GeForce GTX 960**	**Эльбрус-8С**
среднее время выполнения	327,1	73,6	194,6
среднее количество кадров в секунду	3,1	13,6	5,1

Для реализации нейронных сетей с архитектурами VGG19, AlexNet, LeNet, ResNet18, ResNet50, MobileNetV1 с использованием платформы ГНС были получены результаты для процессора Эльбрус-8С. Результаты представлены в табл. 8.

Табл. 8. Время выполнения задачи классификации с платформой ГНС
Table 8. Time to complete the classification task with the GNS platform

		AlexNet	LeNet	ResNet18	ResNet34	ResNet50	VGG19	MobileNetV1
224x224x3	время выполнения (мс)	25,1	12,3	40,6	259,2	316,1	266,7	40,6
	кадров в секунду	39,8	81,2	24,6	3,9	3,2	3,7	24,6
32 x 32 x 3	время выполнения (мс)	5,2	3,3	18,7	27,0	28,4	16,5	4,0
	кадров в	191,3	301,5	53,5	37,1	35,2	60,5	252,1

	секунду							

В табл. 9 представлены результаты производительности некоторых нейронных сетей для вычислителей с использованием современных ускорителей, позволяющие произвести относительную оценку [31]. Для видеокарт использовалась CUDA 5.0.05.

При разнице в количестве ядер в 400 раз и их производительности в 20 раз, производительность на задаче меньше всего в 4 – 15 раз.

Табл. 9. Сравнение практических результатов различных вычислителей
Table 9. Comparison of practical results of various computers

Вычислитель	ядер	fp32 Gflops	дата выхода	ALexNet	VGG-19	ResNet-18	ResNet-34	ResNet-50
GTX 1080	2560	8870	05.2016	7,4	79,8	14,8	24,8	50,8
Maxwell Titan X	3072	6140	03.2015	7,6	93,5	17,1	28,8	56,3
Pascal Titan X	3584	10160	08.2016	5,3	55,8	10,1	16,9	35,0
Эльбрус-8С	8	512	10.2018	25,1	266,7	40,6	259,2	316,1

При моделировании задачи сегментации изображений были получены результаты времени выполнения сегментации видео с разрешением 1280x720 для архитектуры сети ENet.

Среднее время выполнения для одного кадра при разрешении входа нейронной сети 640 x 360 на процессоре Эльбрус-8С оказалось 235 миллисекунд или 4.25 кадра в секунду и 310 миллисекунд или 3.2 кадра в секунду для процессора Intel i7 2600k, для NVidia TX1 практический результат составляет 69 миллисекунд или 14.5 кадра в секунду.

7. Заключение

В ходе данного исследования разработаны математические модели вычислений каскадного классификатора, сверточных нейронных сетей с архитектурой VGG16, VGG19, а также ENet. Получено теоретическое обоснование минимального времени выполнения расчетов на процессорах Эльбрус-2С3, Эльбрус-8С, Эльбрус-8СВ и Эльбрус-16С. Разработаны программные модели для решения задачи обнаружения объектов в видеопотоке с использованием OpenCV. Разработаны программные модели для решения задач классификации с использованием программы на Си, программы на Python, а также с использованием ПО «Платформа-ГНС». Разработаны программные модели для решения задачи сегментации с использованием языка C++.

Проведены эксперименты по обнаружению объектов в видеопотоке с различным разрешением, а также по классификации изображений с разрешением 224 x 224 x 3, 32 x 32 x 3 и сегментации изображений с входом нейронной сети 640 x 360. Результаты, полученные в ходе экспериментов, согласуются с теоретическими результатами, основанными на разработанной математической модели.

В результате проведенных экспериментов показано, что использование микропроцессоров Эльбрус-2С3, Эльбрус-8С, Эльбрус-8СВ и Эльбрус-16С без дополнительных ускорителей обеспечивает достаточную производительность при задаче обнаружения для использования в бортовых вычислителях и системах технического зрения автономных роботов вплоть до разрешения 2560x1440, где обеспечивается частота обработки до 14-15 кадров в секунду. Показано, что обеспечивается производительность до 80 кадров в секунду на задачах классификации изображений 224x224x3 с применением ПО ГосНИИАС. Также показана производительность процессора Эльбрус-8С при решении задачи сегментации изображений в 4.3 кадра в секунду. На основе полученных результатов ожидаемый на практике результат для МП Эльбрус-8СВ составляет более 10 кадров в секунду.

Были продемонстрированы результаты и перспективы использования аппаратно-программной платформы «Эльбрус» для решения задач технического зрения, что позволило согласовать требования к вычислителям перспективных автономных роботов.

Список литературы / References

[1]. Смолин В.С., Соколов С.М. Методы использования искусственного интеллекта в составе систем управления с компьютерным видением. Сборник трудов Всероссийской научно-технической конференции «Техническое зрение в системах управления», 2020 г., стр. 37-38 / Smolin V.S., Sokolov S.M. Methods of using artificial intelligence as part of control systems with computer vision. In Proc. of the All-Russian Scientific and Technical Conference on Technical Vision in Control Systems, 2020, pp. 37-38 (in Russian)

[1] Gondimalla A., Chesnut N. et al. Sparten: A sparse tensor accelerator for convolutional neural networks. In Proc. of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019, pp. 151-165.

[2] Deng L., Li G. et al. Model compression and hardware acceleration for neural networks: A comprehensive survey. Proceedings of the IEEE, vol. 108, issue 4, 2020, pp. 485-532.

[3] Auten A., Tomei M., Kumar R. Hardware acceleration of graph neural networks. In Proc. of the 57th ACM/IEEE Design Automation Conference (DAC), 2020, pp. 1-6.

[4] Ким А.К., Перекатов В.И., Ермаков С.Г. Микропроцессоры и вычислительные комплексы семейства "Эльбрус". СПб, Питер, 2013 г., 272 стр. / Kim A.K., Perekatov V.I., Ermakov S.G. Microprocessors and computer systems of the "Elbrus" family. St. Petersburg, Peter, 2013, 272 p. (in Russian).

[5] Лимонова Е.Е., Бочаров Н.А. и др. Оценка быстродействия системы распознавания на VLIW архитектуре на примере платформы Эльбрус. Программирование, том 45, вып. 1, 2019 г., стр. 15-21 / Limonova E.E., Bocharov N.A. et al. Performance Evaluation of a Recognition System on the VLIW Architecture by the Example of the Elbrus Platform. Programming and Computer Software, vol. 45, issue 1, 2019, pp. 12-17. (in Russian).

[6] Бочаров Н.А., Зуев А.Г., Славин О.А. Производительность микропроцессора Эльбрус-8СВ для решения задач технического зрения в условиях ограничений энергопотребления. Известия Южного федерального университета. Технические науки, вып. 1, 2021 г., стр. 259-271 / Bocharov N.A., Zuev A.G., Slavin O.A. Performance of the Elbrus-8SV microprocessor for technical vision tasks under power constraints. Izvestiya SFedU. Engineering Sciences, issue 1, 2021, pp. 259-271 (in Russian).

[7] Бочаров Н.А., Парамонов Н.Б. и др. Производительность вычислительной техники с процессором «Эльбрус-8С» на задачах робототехнического комплекса. Наноиндустрия, вып. 5(82), 2018 г., стр. 79-84. / Bocharov N.A., Paramonov N.B. et al. Performance of computer systems with Elbrus-8C processor for robotic systems tasks. Nanoindustry, issue 5(82), 2018, pp.79-84 (in Russian).

[8] Кожин А.С. Основные проектные решения для процессора «Эльбрус-16С». Наноиндустрия, том 13, вып. S4, 2020 г., стр. 74-75 / Kozhin A.S. Elbrus-16C processor architecture decisions. Nanoindustrija, vol. 13, №. S4, 2020, pp. 74-75 (in Russian)

[9] Бычков И.Н., Лобанов И.Н., Молчанов И.А. Вычислительная техника на основе аппаратно-программной платформы «Эльбрус» для перспективных информационных систем. Приборы, вып. 8, 2018 г., стр. 14-20 / Bychkov I.N., Lobanov I.N., Molchanov I.A. Computer equipment based on «Elbrus» architecture platform for advanced information systems. Instruments, issue. 8, 2018, pp. 14-20 (in Russian)

[10] Viola P.A., Jones M. Robust real-time object detection. International journal of computer vision, vol. 57, issue 2, 2004, pp. 137-154.

[11] Четверина О.А. Методы коррекции профильной информации в процессе компиляции. Труды ИСП РАН, том 27, вып. 6, 2015 г., стр. 49-68 / Chetverina O.A. Methods of Profile Information Correction during Compilation. Trudy ISP RAN/Proc. ISP RAS, vol. 27, issue 6, 2015, pp. 49-66 (in Russian). DOI: 10.15514/ISPRAS-2015-27(6)-4.

[12] Нейман-заде М.И., Королев С.Д. Руководство по эффективному программированию на платформе «Эльбрус». М., АО «МЦСТ», 2020 г., 178 стр. / Nejman-zade M.I., Korolev S.D. A Guide to Effective Programming on the Elbrus Platform // M., MCST, 2020, 178 p. (in Russian).

[13] Carr S., Guan Y. Unroll-and-jam using uniformly generated sets. In Proc. of 30th Annual International Symposium on Microarchitecture, 1997, pp. 349-357.

[14] Ишин П.А., Логинов В.Е., Васильев П.П. Ускорение вычислений с использованием высокопроизводительных математических и мультимедийных библиотек для архитектуры Эльбрус. Вестник воздушно-космической обороны, вып. 4 (8), 2015 г., стр. 64-68 / Ishin P.A., Loginov V.E., Vasilyev P.P. Computational speed up with use of high efficiency mathematical and multimedia functions. Aerospace Defense Herald, issue 4(8), 2015, pp. 64-68 (in Russian)

[15] Визильтер Ю.В., Желтов С.Ю. Использование глубоких нейронных сетей для анализа данных, управления и оптимизации в перспективных авиационных приложениях. Труды XII мультиконференции по проблемам управления (МКПУ-2019), том 4, 2019 г., стр. 17-20 / Vizilter Yu.V., Zheltov S.Yu. Using deep neural networks for data analysis, management and optimization in promising aviation applications. In Proc. of the XII Multiconference on Problems of Control, 2019, pp. 17-20 (in Russian)

[16] Кожин А.С., Нейман-заде М.И., Тихорский В.В. Влияние подсистемы памяти восьмиядерного микропроцессора «Эльбрус-8С» на его производительность. Вопросы радиоэлектроники, вып. 3, 2017 г., стр. 13-21 / Kozhin A.S., Neiman-zade M.I., Tikhorskiy V.V. Memory subsystem impact on the 8-core «Elbrus-8C» processor performance. Issues of radio electronics, issue 3, 2017, pp. 13-21 (in Russian).

[17] OpenCV Tutorials. Available at: https://docs.opencv.org/3.4/d9/df8/tutorial_root.html, accessed 07.05.2022.

[18] Cascade classifier. Available at: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html, accessed 07.05.2022.

[19] CUDA Toolkit Documentation v10.0.130. Available at: https://docs.nvidia.com/cuda/archive/10.0/, accessed 07.05.2022

[20] Сикорский О.С. Обзор сверточных нейронных сетей для задачи классификации изображений. Новые информационные технологии в автоматизированных системах, вып. 20, 2017 г., стр. 37-42 / Sikorsky O.S. An overview of convolutional neural networks for an image classification problem. New information technologies in automated systems, issue 20, 2017, pp. 37-42 (in Russian)

[21] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556, 2014, 14 p.

[22] Limonova E.E., Neimanzade M.I., Arlazarov V.L. Special aspects of matrix operation implementations for low-precision neural network model on the Elbrus platform. Bulletin of the South Ural StateUniversity. Series Mathematical Modelling, Programming & Computer Software, vol. 13, issue 1, 2020, pp. 118-128.

[23] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. Communications of the ACM, vol. 60, issue 6, 2017, pp 84-90.

[24] Paszke A., Chaurasia A. et al. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147, 2016, 10 p.

[25] Deep Neural Networks (dnn module). Available at: https://docs.opencv.org/4.x/d2/d58/tutorial_table_of_content_dnn.html, accessed 07.05.2022.

[26] ENET CPP. Available at: https://github.com/zm0612/ENet-version-CPP, accessed 07.05.2022.

[27] Semantic Segmentation on PyTorch. Available at:: https://github.com/Tramac/awesome-semantic-segmentation-pytorch, accessed 07.05.2022.

[28] Бирюков А.А., Таранин М.В., Таранин С.В. Процессор 1879ВМ6Я. Реализация глубоких сверточных нейронных сетей. DSPA: Вопросы применения цифровой обработки сигналов, том 8, вып. 4, 2018 г., стр. 191-195 / Birjukov A. A., Taranin M. V., Taranin S. V. Processor 1879VM6Ya. Implementation of deep convolutional neural networks. DSPA: Digital Signal Processing Applications, vol. 8, issue 4, 2018, pp. 191-195 (in Russian).

[29] Петричкович Я. Солохина Т. и др. RoboDeus-50-ядерная гетерогенная СнК для встраиваемых систем и робототехники. Электроника: Наука, технология, бизнес, вып. 7, 2020, pp. 52-63 / Petrichkovich Ya., Solokhina T et al. Robodeus: 50-Core Heterogeneous Soc for Embedded Systems And Robotics. Electronics: Science, Technology, Business, issue 7, 2020, pp. 52-63 (in Russian).

[30] cnn-benchmarks. Available at: https://github.com/jcjohnson/cnn-benchmarks, accessed 07.05.2022.

Информация об авторах / Information about authors

Никита Алексеевич БОЧАРОВ – кандидат технических наук, начальник отдела. Сфера научных интересов: вычислительная техника, техническое зрение, робототехника.

Nikita Alekseevich BOCHAROV – Candidate of Technical Sciences, Head of Department. Research interests: computer engineering, technical vision, robotics.

Николай Борисович ПАРАМОНОВ – доктор технических наук, профессор, главный научный сотрудник. Сфера научных интересов: вычислительная техника, техническое зрение, робототехника.

Nikolay Borisovich PARAMONOV – Doctor of Technical Sciences, Professor, Chief Researcher. Research interests: computer engineering, technical vision, robotics.

Олег Анатольевич СЛАВИН – доктор технических наук, главный научный сотрудник. Сфера научных интересов: вычислительная техника, техническое зрение, распознавание образов.

Oleg Anatolyevich SLAVIN – Doctor of Technical Sciences, Chief Researcher. Research interests: computer engineering, technical vision, document recogntition.

Константин Александрович СУМИНОВ – инженер-программист. Сфера научных интересов: вычислительная техника, техническое зрение, робототехника.

Konstantin Alexandrovich SUMINOV is a software engineer. Research interests: computer engineering, technical vision, robotics.