DOI 10.37882/2223-2966.2024.11.06

МОДЕЛИРОВАНИЕ ПРИМЕНЕНИЯ КОМПЛЕКСНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ДАЛЬНЕМ ГРАНИЧНОМ СЛОЕ ГЕТЕРОГЕННОЙ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

Ветошкин Никита Владимирович
Аспирант, Российская академия народного хозяйства и государственной службы
nikita@vetoshkin.info

Аннотация. Развитие искусственного интеллекта происходит высокими темпами на устройствах с большими вычислительными ресурсами, у которых постоянно увеличивается объем хранения данных и повышается производительность. Однако, внимание к устройствам на периферии представляется низким, ввиду недостатка вычислительных ресурсов. Такие устройства обладают важным преимуществом — большим физическим количеством, что может быть использовано при построении распределенных систем с искусственным интеллектом. В связи с этим, данная статья направлена на выявление возможностей применения периферийных устройств в сложных, в том числе распределенных информационных системах. Проанализированы и описаны три подхода к таким системам: с периферийными устройствами, без них и комплексный подход, когда периферийные устройства формируют вспомогательную роль для сложной системы. Данный подход предполагает внедрение моделей машинного обучения в устройство управления и устройства сбора. Для подхода будет смоделирована ситуация применения в беспилотном транспортном средстве, где объектом детекции выступают дорожные знаки. Для моделируемой ситуации проведен эксперимент, результаты которого подтверждают целесообразность использования комплексного подхода для системы с искусственным интеллектом. По результатам выделены основные преимущества и недостатки внедрения комплексного подхода с использованием периферийных устройств в сложные системы. Данная работа будет полезна исследователям в области интернета вещей, искусственного интеллекта, а также специалистам по разработке беспилотных машин.

Ключевые слова: граничные вычисления, граничный искусственный интеллект, TinyML, распределенные системы, TinyAI, беспилотные транспортные средства, MEC.

MODELING THE APPLICATION OF COMPLEX ARTIFICIAL INTELLIGENCE IN THE FAR EDGE LAYER OF A HETEROGENEOUS DISTRIBUTED SYSTEM

N. Vetoshkin

Summary. The development of artificial intelligence is taking place at a high pace on devices with large computing resources, which are constantly increasing data storage volumes and improving performance. However, attention to devices on the periphery seems to be low due to the lack of computing resources. But such devices also have an important advantage — a large physical quantity, which can be used in building distributed systems with artificial intelligence. In this regard, this article is aimed at identifying the possibilities of using peripheral devices in complex, including distributed information systems. Three approaches to such systems are analyzed and described: with and without peripherals, and an integrated approach when peripherals form an auxiliary role for a complex system. This approach involves the introduction of machine learning models into the control device and the collection device. For the approach, the situation of application in an unmanned vehicle will be simulated, where road signs are the object of detection. An experiment was conducted for the simulated situation, the results of which confirm the expediency of using an integrated approach for a system with artificial intelligence. Based on the results, the main advantages, and disadvantages of implementing an integrated approach using peripheral devices in complex systems are highlighted. This work will be useful for researchers in the field of the Internet of Things, artificial intelligence, as well as specialists in the development of unmanned vehicles.

Keywords: Edge Computing, Edge AI, TinyML, distributed systems, Tiny AI, self-driving cars, MEC.

Введение

С развитием технологий и увеличением количества разнообразных устройств с микроконтроллерами, растет и их вычислительная мощность. Современные микроконтроллеры обладают высокой производительностью и низким энергопотреблением, что делает их привлекательными для использования в различных сферах [6].

Однако, в отрасли информационных технологий наблюдается переход от облачных вычислений к граничным вычислениям [1][2]. Это связано с несколькими ключевыми причинами. Во-первых, с ростом числа устройств, подключенных к сети интернет, увеличивается нагрузка на сетевую инфраструктуру. Во-вторых, вопросы безопасности данных и государственное регулирование требуют хранения конфиденциальной информации на собственной инфраструктуре. В-третьих, в некоторых сферах, таких как сельское хозяйство, добыча полезных ископаемых и энергетика, требуется высокая отказоустойчивость и автономность работы граничной системы. Наконец, сокращение эксплуатационных расходов на поддержание систем становится все более актуальным [4][9].

В связи с этим, архитектура распределенных систем претерпевает изменения. Она включает в себя три основных слоя: облачный слой, ближний граничный слой и дальний граничный слой [3][11]. Ближний граничный слой содержит сетевую инфраструктуру и телекоммуникационное оборудование, а дальний граничный слой — устройства, применяемые в рамках граничных вычислений. Для последнего слоя разрабатываются концепции архитектур, такие как туманные вычисления, Multi-Access Edge Computing (MEC) и Cloudlet [5][7][10].

Устройства в дальнем граничном слое можно разделить на три группы: высокопроизводительные устройства (сервера, кластеры, графические ускорители), устройства с микроконтроллерами и устройства с микропроцессорами. Задача устройств с микроконтроллерами — сбор данных, а устройств с микропроцессорами — обработка и сведение данных. Последние обладают достаточной производительностью для работы с моделями искусственного интеллекта [8].

В данной работе будет проведен анализ применения комплексного искусственного интеллекта в рамках дальнего граничного слоя. Будет воссоздан подход по созданию нового разделения вычислительных ресурсов для граничных вычислений, где предобработка данных осуществляется на устройствах с микроконтроллером, а результаты передаются на устройства с микропроцессором. Осуществлен эксперимент, где модель машинного обучения будет запущена на устройстве с микропроцессором и на устройствах с микроконтроллером. По результатам будут сформированы выводы о возможной эффективности переноса легковесных моделей машинного обучения на устройства с микроконтроллерами.

Материалы и методы

Выделяют 3 основных слоя распределенных систем с применением граничных вычислений: облачный слой, ближний граничный слой, дальний граничный слой [1][3]. Пример такого разделения на слои и категории устройств, которые используются в этих слоях представлен на рисунке 1.

Понятие Far Edge (дальний граничный слой) было впервые описано Китом Бэзилом в 2021 году [11]. Дальний граничный слой определяется как сетевое и инфраструктурное пространство, которое эксплуатируется и принадлежит организациям, являющимся конечным пользователем системы. Слой имеет множество секторов применения: коммерция, промышленность, государство. Количество устройств в таком слое может достигать десятки тысяч. Основной функцией слоя является агрегация и анализ данных [4][9].

В данной статье рассматриваются системы имеющие в составе 2 группы устройств: устройства с микропроцессором (микрокомпьютеры, SoC) и устройства с микроконтроллером [6][8].

Размещение моделей машинного обучения на одноплатных компьютерах (SoC) на базе Linux не является сложным процессом в виду единого интерфейсного входа — ядра Linux. Доступные сегодня 32-разрядные микроконтроллеры имеют гораздо меньшую стоимость по сравнению с микрокомпьютерами, и для их работы требуется всего несколько милливатт, что делает их энергоэффективной и экономичной альтернативой [12]. Однако такие устройства имеют строгие ограничения ресурсов и содержат встроенные операционные системы реального времени не на базе Linux и множество

*МК-микроконтроллер

Рис. 1. Компоненты распределенных систем с применением граничных вычислений

других подобных ограничений, что порождает разнородность внедрения легковесных моделей искусственного интеллекта [13].

Моделирование ситуации в гетерогенной распределенной системе

Для проведения эксперимента требуется моделирование потенциального сценария использования комплексного искусственного интеллекта в рамках дальнего граничного слоя. В качестве системы на границе был выбран беспилотный наземный транспорт [14].

Выбор беспилотного транспорта обоснован тем, что на граничный микрокомпьютер (SoC) поступает большое количество потоков данных с датчиков для последующей обработки и принятия решений. Частота передачи экземпляров данных в потоках является высокой. Датчики представляют из себя различные устройства сбора, в том числе под управлением микроконтроллера [15][16].

В моделируемой ситуации в качестве устройства управления выступает граничный микрокомпьютер, а устройствами для сбора данных — набор камер под управлением микроконтроллеров на транспортном средстве [17].

В качестве объекта детекции выступает набор дорожных знаков.

Можно выделить три потенциальных варианта использования искусственного интеллекта для обнаружения дорожных знаков на изображении в беспилотном транспортном средстве.

Первый подход предполагает устройство управления с искусственным интеллектом, которое находится на удаленном сервере — в облаке. Реализация имеет существенный недостаток — нестабильность работы с сетью ввиду разнородности географического применения системы [18].

Второй подход является граничным решением, где искусственный интеллект запускается на микрокомпьютере SoC (System-on-Chip). С устройств сбора на микрокомпьютер поступает видеопоток данных. Таких потоков может быть несколько и их количество зависит от набора устройств в рассматриваемой системе [19].

Третий подход является моделируемым. Искусственный интеллект запускается на микрокомпьютере и микроконтроллерах. Обработка видеопотока осуществляется на устройствах сбора данных. Микроконтроллер является актуатором сценариев последующей обработки для устройств управления на базе микропроцессоров.

Применяя теорию систем для второго подхода, устройством управления служит микрокомпьютер SoC,

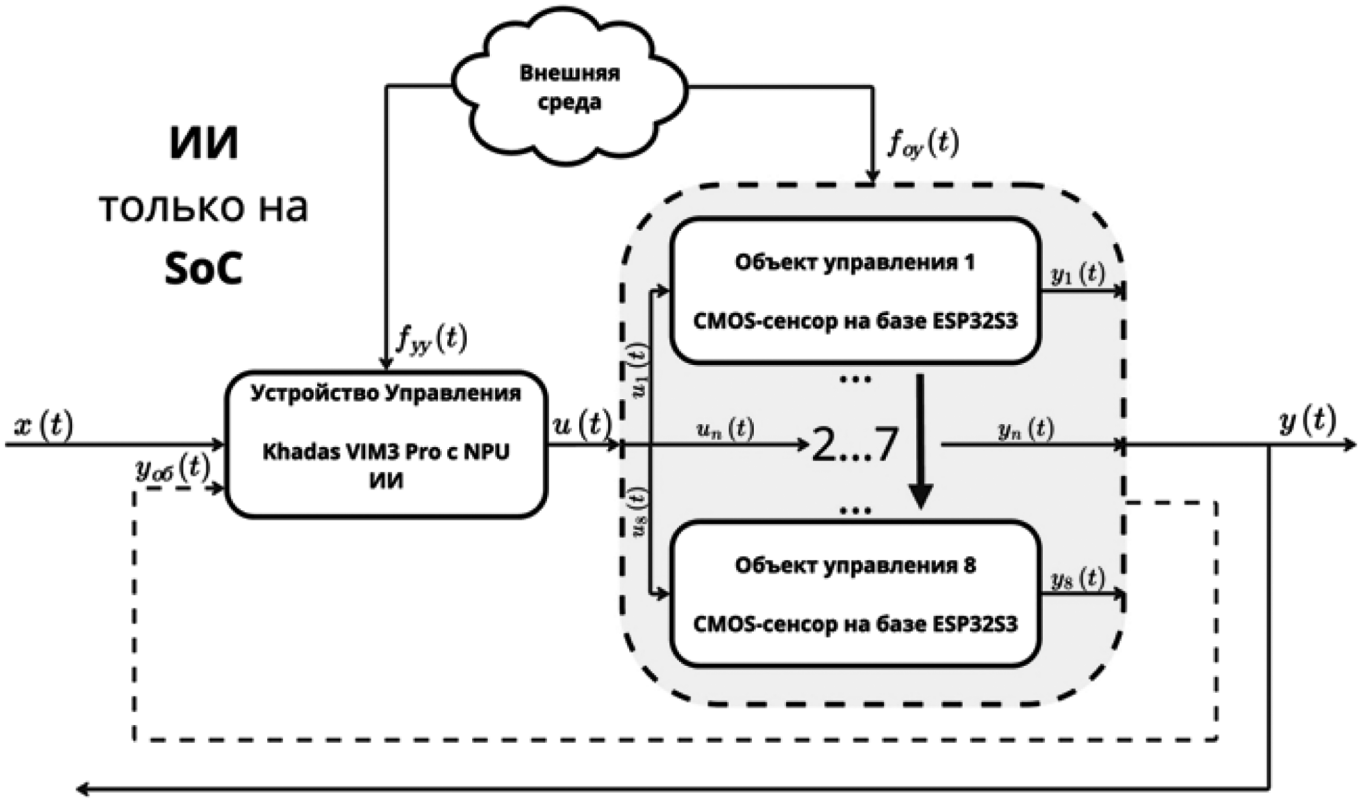

Рис. 2. Схема концепции архитектуры системы с одним уровнем устройств управления

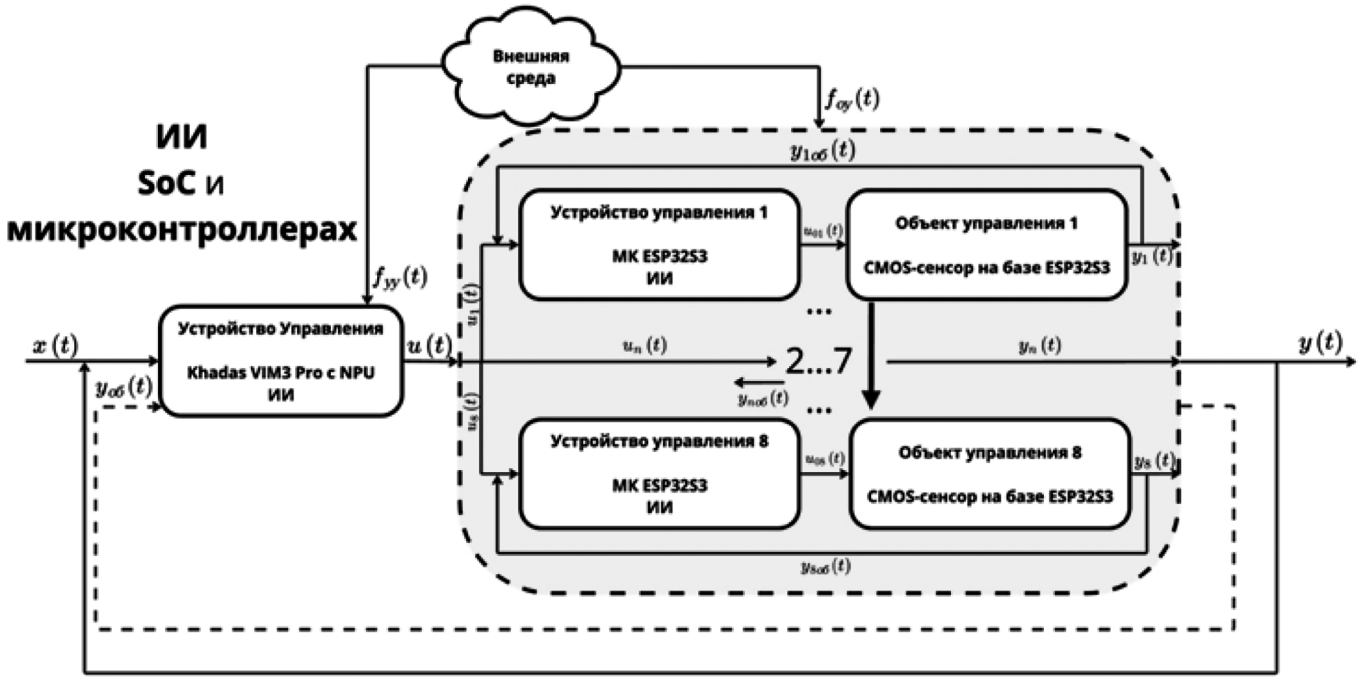

Рис. 3. Схема концепции архитектуры с двумя уровнями устройств управления

объектом управления — набор камер. Схема реализации представлена на рисунке 2.

Моделируемое решение содержит устройство общего управления — микрокомпьютер SoC, дополнительные устройства управления в виде микроконтроллеров, взаимодействующих с CMOS сенсором. Схема моделируемого решения представлена на рисунке 3.

Оборудование и материалы для проведения эксперимента

Оборудование для проведения эксперимента:

SoC Khadas VIM3 Pro с нейропроцессиновыми ядрами (NPU), ключевые технические характеристики представлены в таблице 1.

Таблица 1.

Основные технические характеристики микрокомпьютера Khadas VIM3 Pro

Показатель	Характеристика
SoC	Amlogic A311D
CPU	4× ARM Cortex-A73 @ 2.2 ГГц 2× ARM Cortex-A53 @1.8 ГГц
GPU	ARM Mali-G52 MP4 @800 МГц
RAM	4 Гб LPDDR4
NPU	5 TOPS (количество операций в секунду в триллионах)

Устройства сбора с микроконтроллером ESP32S3 в количестве 8 штук. Технические характеристики представлены в таблице 2.

Таблица 2.

Основные технические характеристики отладочной платы с микроконтроллером ESP32S3

Показатель	Характеристика
CPU	2× 32-bit LX7 @ 240МГц
ROM	384 Кб
SRAM	512 Кб
PSRAM	8 Мб
FLASH	16 Мб

На рисунке 4, сверху — микрокомпьютер Khadas VIM3 Pro, снизу — 8 отладочных плат с микроконтроллером ESP32S3.

Для проведения эксперимента была использована предобученная модель tensorflow lite, задачей которой является определение дорожных знаков на изображении размерностью 48x48 пикселей в градациях серого. Над моделью предварительно был проведен процесс квантования, входной и выходной тензор является целочисленным.

На вход модели поступает тензор, размерностью (1x48x48x1), на выходе массив с 4 классами дорожных знаков: только поворот налево, запрет обгона, только поворот направо, стоп.

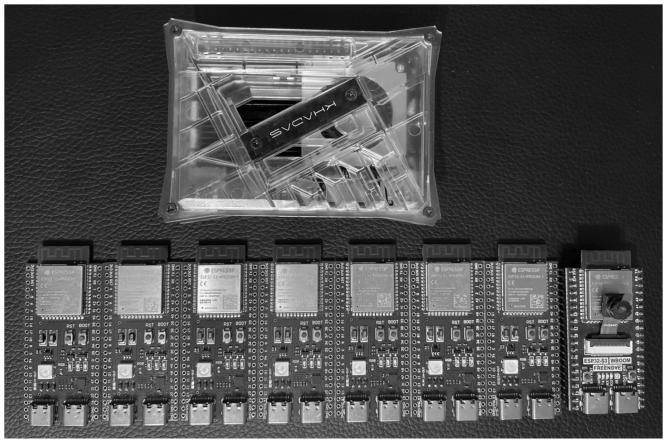

Рис. 4. Состав оборудования для проведения эксперимента

Тестовый набор содержит 1011 изображений, содержащих дорожные знаки. Примеры таких изображений представлены на рисунке 5.

Результаты

Для архитектуры без управляющих устройств с микроконтроллером, было проведено 10 итераций работы модели. Модель запускалась на микрокомпьютере (KHADAS VIM3 Pro) с применением ядер NPU. На рисунке 6 представлена зависимость количества обработанных изображений от накопленной суммы времени работы по обработке изображений искусственным интеллектом. Время, учитываемое в накопленной сумме, характеризует разницу между вводом входного тензора и получением выходного тензора с результатами. В эксперименте не учитывается время на передачу, предобработку и постобработку данных.

Ключевые результаты 10 итераций представлены в таблице 3.

Для архитектуры, содержащей управляющие устройства с микроконтролером, было проведено несколько итераций с различным количеством устройств сбора. Модель была загружена в микроконтроллеры ESP32S3. Для аппаратного ускорения работы модели использовалась специализированная библиотека ESPNN. Результаты итераций представлены на рисунке 7 и в таблице 4.

На рисунке 8 прослеживается тренд уменьшения времени по гиперболическому закону. С увеличением числа используемых устройств сбора время выполнения уменьшается, вплоть до времени выполнения обработки одного изображения на SoC.

Таким образом, при подключении большого количества устройств сбора, целесообразно переносить работу модели искусственного интеллекта непосредственно на устройство управления с микроконтроллером. Дополнительно, можно сделать вывод о том, что на устройства управления с микроконтроллером рекомендуется выносить легковесные нейросети с задачей генерации событий-актуаторов для всей системы, в случае обнаружения необходимых данных.

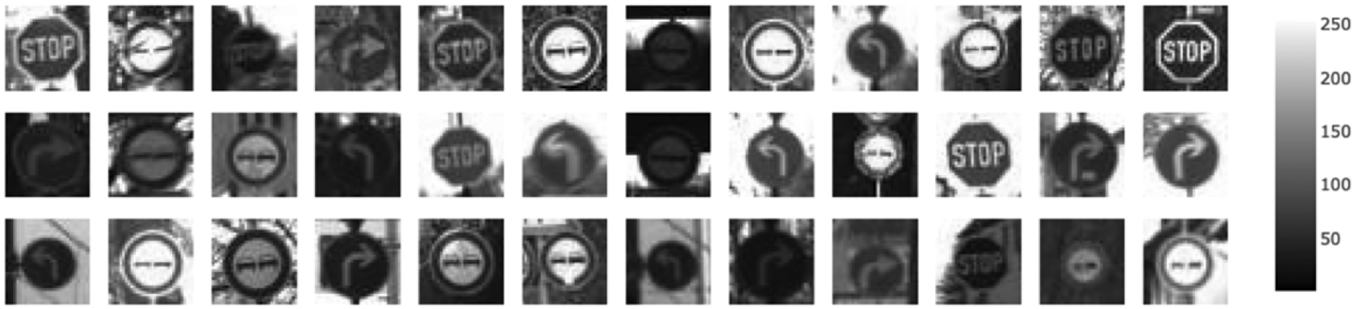

Рис. 5. Примеры изображений, содержащих дорожные знаки из тестового набора

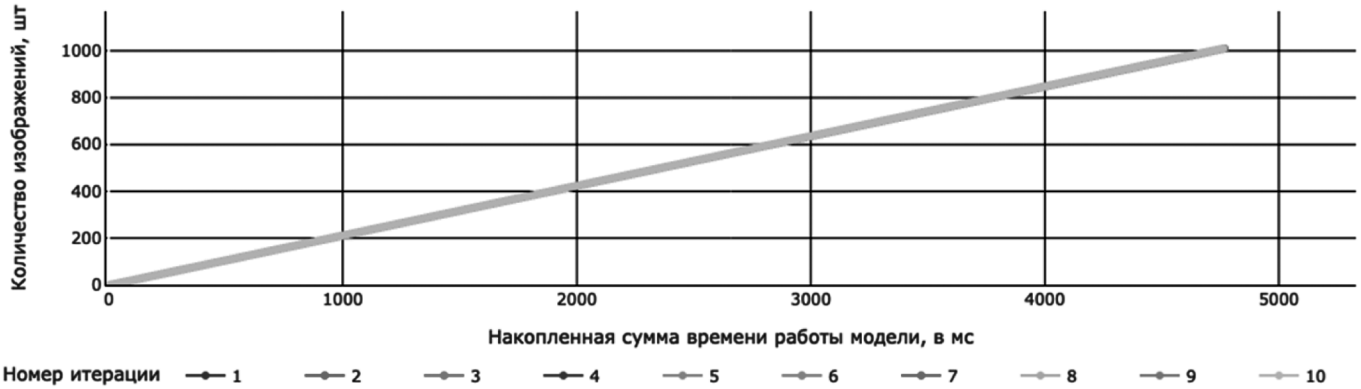

Рис. 6. График накопленного времени по обработке моделью изображений на Khadas VIM3 Pro

Таблица 3.

Результаты работы модели на микрокомпьютере Khadas VIM3 Pro

№ итерации	Минимальное время обработки в мс	Медианное время обработки в мс	Среднее время обработки в мс	Максимальное время обработки в мс	Время обработки всего тестового набора в мс
1	4.684905	4.70653	4.712795179	5.034448	4764.635926
2	4.688238	4.707446	4.7143036508	5.060865	4766.160991
3	4.681071	4.708613	4.7168851444	5.107491	4768.770881
4	4.683697	4.706363	5.3198838991	616.58282	5378.4026219
5	4.68353	4.707405	4.7145564015	5.027282	4766.416522
6	4.687072	4.708405	4.7158674758	5.024115	4767.742018
7	4.684988	4.703363	4.709582453	5.124449	4761.38786
8	4.676321	4.704696	4.71099111276	5.056657	4762.812015
9	4.685071	4.706405	4.7120031691	4.834697	4763.835204
10	4.676905	4.705571	4.7120400553	5.160991	4763.872496

Таблица 4.

Результаты работы модели в распределённой системе с ESP32S3.

Количество узлов распределённой системы	Минимальное время обработки в мс	Медианное время обработки в мс	Среднее время обработки в мс	Максимальное время обработки в мс	Время обработки всего тестового набора в мс
1	90.422	92.671	92.602874382	95.305	93621.506
2	90.5935	92.08525	93.679438735	223.4685	47401.796
4	90.6055	91.072	92.182599802	109.64075	23322.19775
6	90.6227	91.424	93.906321499	137.0092	15870.1683
8	90.68275	91.175	93.690965879	109.264875	11898.7527

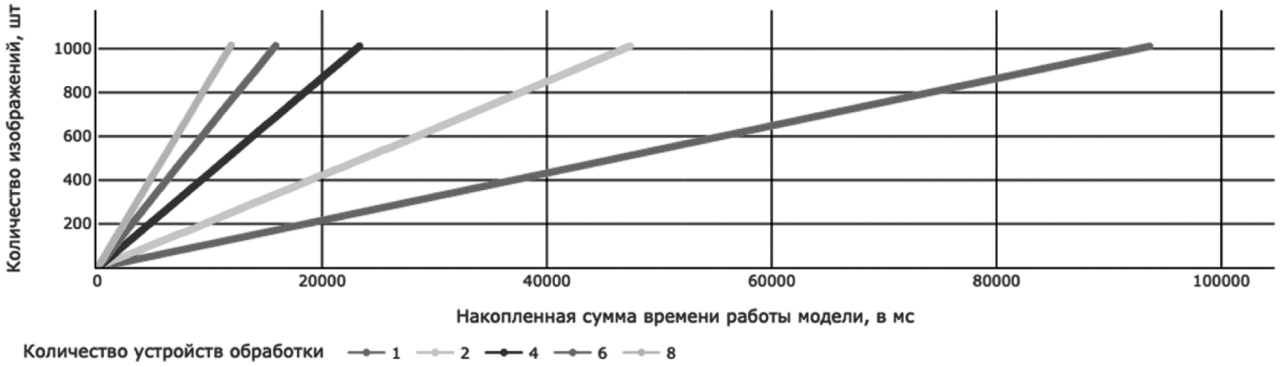

Рис. 7. График накопленного времени по обработке моделью изображений в зависимости от количества подключенных ESP32S3 к распределенной системе

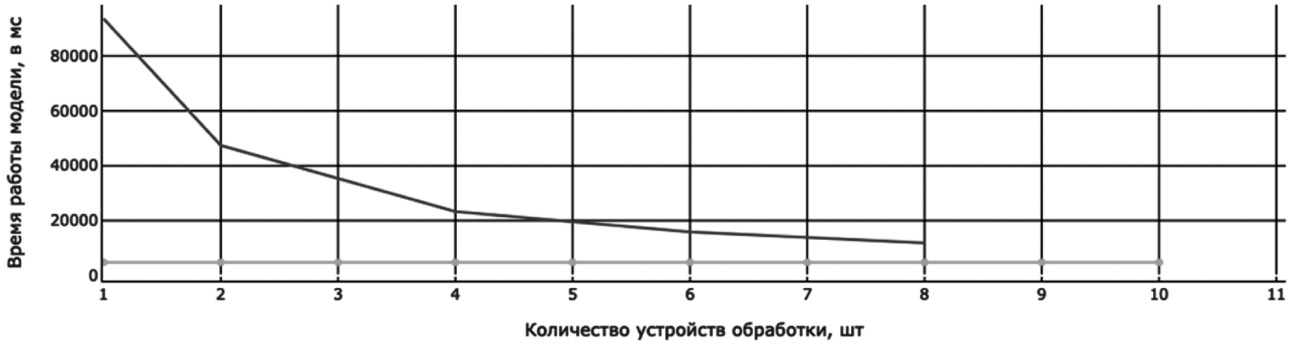

Рис. 8. График накопленного времени по обработке моделью изображений в зависимости от количества подключенных ESP32S3 к распределенной системе и ограничение с обработкой только на микрокомпьютере SoC

В результате эксперимента можно выделить следующие преимущества такого подхода:

1. Устройство управления с микроконтроллером имеет функцию актуатора для событий последующей обработки на устройстве сбора — микрокомпьютере. Таким образом, линии связи для обмена данными между устройствами менее загружены.
2. Снижение общего энергопотребления системы. В рамках работы системы снижается нагрузка на микрокомпьютер (SoC). В результате чего снижается энергопотребление всей системы.

Заключение

В этой статье была представлена система применения комплексного искусственного интеллекта относительно дальнего граничного слоя распределенной системы. Внедрение искусственного интеллекта в микрокомпьютеры и микроконтроллеры в рамках такой системы позволяет:

1. Перераспределить вычислительную нагрузку системы и обеспечить ее снижение на основное устройство управления.
2. Снизить энергопотребление всей системы
3. Использовать ресурсы устройства управления по требованию. При обнаружении устройством сбора с искусственным интеллектом необходимых данных, генерируется событие для активации ресурсов общего устройства управления.
4. Снизить поток данных межмашинного взаимодействия в рамках системы. Снижается нагрузка на каналы связи между устройствами, ввиду снижения частоты обмена данными

Недостатком использования такой системы служит разнородность устройств сбора и их архитектур. Не существует единого интерфейса для работы моделей машинного обучения на различных типах микроконтроллеров.

ЛИТЕРАТУРА

1. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge Computing: Vision and Challenges // IEEE Internet of Things Journal. 2016. Vol. 3, № 5. P. 637–646.
2. Satyanarayanan M. The Emergence of Edge Computing // Computer. 2017. Vol. 50, № 1. P. 30–39.
3. Bonomi F., Milito R., Zhu J., Addepalli S. Fog Computing, and Its Role on the Internet of Things // Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. 2012. P. 13–16.
4. Mao Y., You C., Zhang J., Huang K., Letaief K.B. A Survey on Mobile Edge Computing: The Communication Perspective // IEEE Communications Surveys & Tutorials. 2017. Vol. 19, № 4. P. 2322–2358.
5. Taleb T., Samdanis K., Mada B., Flinck H., Dutta S., Sabella D. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration // IEEE Communications Surveys & Tutorials. 2017. Vol. 19, № 3. P. 1657–1681.
6. Lin J., Yu W., Zhang N., Yang X., Zhang H., Zhao W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications // IEEE Internet of Things Journal. 2017. Vol. 4, № 5. P. 1125–1142.
7. Dastjerdi A.V., Buyya R. Fog Computing: Helping the Internet of Things Realize Its Potential // Computer. 2016. Vol. 49, № 8. P. 112–116.
8. Chen X., Jiao L., Li W., Fu X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing // IEEE/ACM Transactions on Networking. 2016. Vol. 24, № 5. P. 2795–2808.
9. Wang S., Zhang X., Zhang Y., Wang L., Yang J., Wang W.A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications // IEEE Access. 2017. Vol. 5. P. 6757–6779.
10. Chiang M., Zhang T. Fog and IoT: An Overview of Research Opportunities // IEEE Internet of Things Journal. 2016. Vol. 3, № 6. P. 854–864.
11. Basil K. Near, Far or Tiny: Defining and Managing Edge Computing in a Cloud Native World. VMBlog, 27 Apr. 2021, https://vmblog.com/archive/2021/04/27/near-far-or-tiny-defining-and-managing-edge-computing-in-a-cloud-native-world.aspx
12. Banbury C.R., Reddi V.J., Lam M., et al. Benchmarking TinyML Systems: Challenges and Direction // arXiv preprint arXiv:2003.04821. 2020.
13. David R., Duke J., Jain A., et al. TensorFlow Lite Micro: Embedded Machine learning on TinyML Systems // arXiv preprint arXiv:2010.08678. 2020.
14. Grigorescu S., Trasnea B., Cocias T., Macesanu G. A survey of deep learning techniques for autonomous driving // Journal of Field Robotics. 2020. Vol. 37, № 3. P. 362–386.
15. Zhu L., Yu F.R., Wang Y., Ning B., Tang T. Big Data Analytics in Intelligent Transportation Systems: A Survey // IEEE Transactions on Intelligent Transportation Systems. 2019. Vol. 20, № 1. P. 383–398.
16. Mahmud R., Koch F.L., Buyya R. Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions // Proceedings of the 19th International Conference on Distributed Computing and Networking. 2018. P. 1–10.
17. Teichmann M., Weber M., Zöllner M., Cipolla R., Urtasun R. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving // 2018 IEEE Intelligent Vehicles Symposium (IV). 2018. P. 1013–1020.
18. Zhang W., Zhang Z., Chao H.-C. Cooperative Fog Computing for Dealing with Big Data on the Internet of Vehicles: Architecture and Hierarchical Resource Management // IEEE Communications Magazine. 2017. Vol. 55, № 12. P. 60–67.
19. Kang Y., Hauswald J., Gao C., et al. Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge // Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems. 2017. P. 615–629.

Журнал «Современная наука: актуальные проблемы теории и практики»