

ОЦЕНКА БЫСТРОДЕЙСТВИЯ СИСТЕМЫ РАСПОЗНАВАНИЯ НА VLIW АРХИТЕКТУРЕ НА ПРИМЕРЕ ПЛАТФОРМЫ ЭЛЬБРУС¹

© 2019 г. Е. Е. Лимонова^{a,b,*}, Н. А. Бочаров^{c,**}, Н. Б. Парамонов^{d,***},
Д. С. Богданов^{a,****}, В. В. Арлазаров^{a,b,*****}, О. А. Славин^{a,b,*****}, Д. П. Николаев^{b,e,*****}

^aИнститут системного анализа Федерального исследовательского центра “Информатика и управление” РАН
117312 Москва, пр-т 60-летия Октября, 9, Россия

^bSmart Engines 117312 Москва, пр-т 60-летия Октября, 9, Россия

^cАО МЦСТ 119334 Москва, ул. Вавилова, 24, Россия

^dПАО “ИНЭУМ им. И.С. Брука” 119334 Москва, ул. Вавилова, 24, Россия

^eИнститут проблем передачи информации им. А.А. Харкевича РАН
127051 Москва, Большой Каретный переулок, д. 19 стр. 1, Россия

* e-mail: limonova@smartengines.biz

** e-mail: bocharov.na@phystech.edu

*** e-mail: paramonov_n_b@mail.ru

**** e-mail: dsbogdanov@yahoo.com

***** e-mail: vva777@gmail.com

***** e-mail: oslavin@isa.ru

***** e-mail: dimonstr@iitp.ru

Поступила в редакцию 01.06.2018 г.

После доработки 01.06.2018 г.

Принята к публикации 12.06.2018 г.

В работе содержится обзор современных вычислительных устройств на основе VLIW-архитектуры Эльбрус, а также выполняется экспериментальная проверка работоспособности и оценка быстродействия системы распознавания документов Smart IDReader на этих устройствах. Описаны методы повышения быстродействия распознающей системы на платформе Эльбрус, приведены экспериментальные оценки полученных ускорений, а также проведено сравнение производительности различных устройств на основе процессоров Эльбрус.

DOI: 10.1134/S0132347419010047

1. ВВЕДЕНИЕ

На сегодняшний день по-прежнему наблюдается рост производительности вычислительных устройств, как предназначенных для решения конкретных задач, так и общего назначения. Однако решаемые задачи также усложняются, и к вычислительным устройствам предъявляются новые требования. Часто современные алгоритмы должны работать в режиме жестких временных ограничений, как, например, в случае обнаружения объектов в режиме реального времени [1, 2].

Таким образом, задача оценки и оптимизации быстродействия приложений при наличии ограничений времени работы приобретает большое практическое значение. К таким приложениям относятся и системы массового ввода документов

[3, 4]. Для подобных систем время работы является крайне важным параметром, поскольку зачастую они являются частью некоторой системы массового обслуживания, и время распознавания оказывает прямое влияние на эффективность всей системы.

Одним из методов повышения быстродействия является использование программируемых логических интегральных схем для реализации всей системы или отдельных ее модулей [5, 6], однако это довольно сложный и трудоемкий процесс, подходящий для создания программно-аппаратных платформ. Поэтому немало работ посвящено повышению быстродействия отдельных алгоритмов и методов на платформах общего назначения, например, на мобильных устройствах [7–9]. В этом случае для оптимизации могут использоваться свойства конкретной платформы, такие, как наличие SIMD-расширений или не-

¹ Работа выполнена при поддержке РФФИ (проекты № 17-29-03170, 17-29-03297 и 18-07-01384).

скольких ядер, и другие особенности архитектуры вычислителя.

Данная работа посвящена оценке максимального быстродействия системы массового ввода документов Smart IDReader на VLIW-архитектуре на примере платформы Эльбрус. Одним из этапов такой оценки является оптимизация системы с учетом особенностей данной вычислительной архитектуры.

2. АРХИТЕКТУРА ЭЛЬБРУС

Архитектура Эльбрус относится к архитектурам, использующим принцип широкого командного слова (Very Long Instruction Word, VLIW). При генерации исполняемого кода для процессоров с VLIW-архитектурой компилятор формирует последовательности групп команд (широкие командные слова), в которых отсутствуют зависимости между командами внутри каждой группы и сведены к минимуму зависимости между командами в разных группах. При исполнении команды каждой группы запускаются параллельно, что обеспечивает высокий уровень параллелизма на уровне команд [10]. Такое распараллеливание на уровне команд целиком обеспечивается оптимизирующим компилятором, разработанным с учетом особенностей конкретной архитектуры. В случае с архитектурой Эльбрус, это оптимизирующий компилятор lcc.

Другой особенностью процессоров архитектуры Эльбрус являются методы работы с памятью. Помимо наличия кэша, позволяющего оптимизировать время доступа в память, ими поддерживаются методы предварительной подкачки данных, которые позволяют прогнозировать обращения в память и производить подкачку данных в кэш или другое специальное устройство за некоторое время до их использования. Процессоры архитектуры Эльбрус поддерживают программно-аппаратный метод подкачки. Это означает, что аппаратная часть микропроцессора включает в себя специальное устройство для обращения к массивам (Array Access Unit, AAU), в то время как необходимость подкачки определяется компилятором, генерирующим специальные инструкции для AAU. Использование устройства подкачки эффективнее помещения элементов массива в кэш, поскольку элементы массивов чаще всего обрабатываются последовательно и редко используются более одного раза [11]. Однако необходимо отметить, что использование буфера предварительной подкачки на Эльбрусе возможно только при работе с выровненными данными. За счет этого чтение/запись выровненных данных происходят заметно быстрее, чем соответствующие операции для невыровненных данных.

Также микропроцессоры Эльбрус поддерживают несколько видов параллелизма помимо параллелизма на уровне команд: векторный параллелизм [12], параллелизм потоков управления, параллелизм задач в многомашинном комплексе.

3. СИСТЕМА РАСПОЗНАВАНИЯ ДОКУМЕНТОВ

Система распознавания документов должна обеспечивать последовательность выполнения следующих этапов обработки изображения и нескольких изображений документа в видеопоследовательности:

1. Оценка качества входного изображения, обработка и нормализация изображения [13–15].
2. Поиск документа на изображении: данный этап включает в себя алгоритмы на основе анализа локальных особенностей изображений и модифицированного детектора Виолы и Джонса [16, 17].
3. Определение конкретного типа документа на изображении [18, 19].
4. Поиск отдельных полей документа. Это могут быть как поля, содержащие текстовую информацию, так и фотография или подпись [20, 21].
5. Сегментация и распознавание текстовых полей, выполняемые с помощью алгоритмов динамического программирования и оптимизированных нейронных сетей [22–26] и аналогичных методов распознавания символов [27, 28].
6. Обработка полученной информации: уточнение результатов распознавания [29–31]. Объединение результатов распознавания нескольких кадров с учетом их качества [32, 33] в случае распознавания видеопоследовательностей.

Все эти этапы реализуются в рамках системы распознавания Smart IDReader [34, 35]. Эта система позволяет распознавать документы, удостоверяющие личность, на различных платформах. Доступные режимы распознавания позволяют извлекать данные держателя документа из последовательности кадров видеоролика, цифровых фото или отсканированных изображений документов.

Очевидно, что общее время работы системы зависит от времени работы каждого этапа. В свою очередь, например, время поиска документа зависит от положения документа на изображении и того, насколько он повернут. Определение типа документа также может занимать разное время при разном возможном числе типов. Поэтому в данной работе для оценки производительности распознающей системы использовалось общее время распознавания, т.е. суммарное время работы всех модулей системы.

Система Smart IDReader может работать в клиентском и серверном режимах. Клиентский режим предполагает, что система работает на

устройстве конечного пользователя, который выполняет сканирование, фото- или видеосъемку документа и затем запускает распознавание полученных изображений на своем устройстве. При этом такое устройство может обладать весьма ограниченной вычислительной мощностью. Поэтому в клиентском режиме распознавание изображений нескольких разных документов выполняется последовательно. При этом в рамках каждого отдельного запуска распознавания некоторые части системы были распараллелены.

Серверный режим предполагает, что система работает на многоядерном устройстве достаточно большой вычислительной мощностью, на котором можно параллельно запустить несколько независимых процессов распознавания документов. В таком режиме система полностью загружает устройство и демонстрирует максимальную производительность.

4. СПОСОБЫ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ РАСПОЗНАВАНИЯ НА ПЛАТФОРМЕ ЭЛЬБРУС

В нашей работе использовались две технологии для повышения быстродействия распознающей системы: распараллеливание вычислений и реализации функций низкоуровневой обработки изображений с помощью интринсиков.

Распараллеливание вычислений выполнялось на максимально доступное число потоков при помощи библиотеки `tbb`. Для этого использовались стандартные методы из библиотеки `tbb`, такие как `parallel_for`, позволяющий распараллелить цикл `for`, и `task_group`, позволяющий создать набор из нескольких независимых задач, которые затем могут исполняться параллельно.

Были распараллелены этапы поиска документа на изображении (на основе алгоритма Виолы и Джонса), определения конкретного типа документа на изображении (при помощи сравнения документа с шаблоном с помощью расширенного алгоритма RANSAC), поиска текстовых полей (выполняется независимо в каждой из частей документа), а также сегментации и распознавания текстовых полей на основе нейронных сетей. В случае алгоритма Виолы и Джонса и метода определения типа документа удалось выполнить распараллеливание циклов, поскольку действия на каждой итерации цикла независимы. Поиск текстовых полей также выполняется независимо в каждой из частей документа, и может быть легко распараллелен. Сегментация и распознавание символов работают для каждого найденного поля независимо и также были распараллелены стандартными методами библиотеки `tbb`.

Интринсики — функции, вызовы которых замечаются компилятором на высокоэффективный

код для данной платформы. С помощью интринсиков разработчики могут использовать векторный параллелизм: выполнять одну и ту же операцию над одним регистром, содержащим сразу несколько элементов данных. Микропроцессоры Эльбрус-4С и Эльбрус-8С поддерживают набор интринсиков, для которого размер регистра составляет 64 бита. Он включает в себя операции для преобразования данных, инициализации элементов вектора, арифметических операций, побитовых логических операций, перестановки элементов вектора и др. Кроме того, для процессоров Эльбрус доступна библиотека EML, содержащая эффективные реализации основных функций обработки данных.

При использовании интринсиков на платформе Эльбрус следует уделять особое внимание доступу в память, поскольку в задачах обработки изображений часто требуется невыровненное чтение данных в 64-битном регистре. Такое чтение само по себе неэффективно, так как требует пары команд чтения и последующей команды формирования блока данных, но, что еще важнее, при этом не может использоваться буфер подкачки массивов, повышающий скорость доступа к данным на платформе Эльбрус. Таким образом, низкоуровневую обработку данных, например, числовых массивов, следует выполнять в несколько этапов: обработка начальной части (до границы выравнивания на 64-бита), обработка основной части, использующая выровненный доступ к памяти, и обработка оставшихся элементов массива. Поскольку анализ указателей во время компиляции является нетривиальной задачей, можно использовать флаг компилятора — `faligned`, с которым все операции доступа к памяти выполняются выровненным образом.

Следующая особенность использования интринсиков на платформе Эльбрус связана непосредственно с VLIW-архитектурой. Благодаря наличию нескольких арифметико-логических устройств (АЛУ), которые работают параллельно и загружаются при формировании широких командных слов, несколько команд могут исполняться одновременно. Процессоры Эльбрус-4С и Эльбрус-8С содержат шесть АЛУ, которые можно задействовать в рамках одной широкой команды, однако каждое АЛУ поддерживает свой набор интринсиков. Простые операции, например, сложение или умножение элементов в 64-битных регистрах, как правило, поддерживаются двумя АЛУ. Это означает, что процессор Эльбрус может исполнить по две таких инструкции за один такт. Для этого в исполняемом коде следует использовать развертывание циклов. Компилятор `lcc` поддерживает программу `#pragma unroll(n)`, которая позволяет выполнить развертывание `n` итераций цикла.

Таблица 1. Характеристики тестируемых машин с архитектурой Эльбрус

Машина	Эльбрус 101-РС	Эльбрус 401-РС	Эльбрус-4.4	Эльбрус 801-РС	Эльбрус-8.4
Процессор	Эльбрус-1С+	Эльбрус-4С	Эльбрус-4С	Эльбрус-8С	Эльбрус-8С
Ядра общего назначения	1	4	16	8	32
Тактовая частота, МГц	985	800	750	1200	1200
Операции за такт (на ядро)	до 25	до 23	до 23	до 25	до 25
Объем ОЗУ	16 ГБ	24 ГБ	96 ГБ	32 ГБ	128 ГБ
Ширина SIMD-регистра	64 бита	64 бита	64 бита	64 бита	64 бита
L1 кэш (на ядро)	64 КБ	64 КБ	64 КБ	64 КБ	64 КБ
	данные + + 128 КБ команды	данные + + 128 КБ команды	данные + + 128 КБ команды	данные + + 128 КБ команды	данные + + 128 КБ команды
L2 кэш (на ядро)	2 МБ	2 МБ	2 МБ	512 КБ	512 КБ
L3 кэш (общая)	—	—	—	16 МБ	64 МБ

Таким образом, для эффективного использования интринсиков на платформе Эльбрус необходимо:

- обеспечить выровненный доступ в память;
- обеспечить использование всех доступных АЛУ путем развертывания циклов.

В нашей работе с помощью функций EML и интринсиков был реализован ряд операций, применяемых при низкоуровневой обработке изображений и распознавании образов:

- арифметические операции под отдельными пикселями изображения;
- масштабирование изображения;
- транспонирование изображения;
- поворот изображения;
- фильтрация изображения;
- умножение и сложение матриц вещественных чисел.

5. ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ РАСПОЗНАЮЩЕЙ СИСТЕМЫ

Рассмотренная система распознавания документов была скомпилирована из исходного кода с

Таблица 2. Среднее время распознавания паспорта РФ на Эльбрус-401РС с различными методами повышения быстродействия

Вид оптимизации	Среднее время распознавания, сек
Без оптимизации	10.01
С распараллеливанием без интринсиков	4.22
С распараллеливанием и интринсиками	1.90

помощью lcc [36] версии 1.21.19 и работала в нативном режиме. Распараллеливание выполнялось на максимально доступное число потоков при помощи библиотеки tbb [37]. Библиотека tbb является кроссплатформенной и была собрана lcc из исходного кода для процессоров семейства Эльбрус.

Далее распознающая система была запущена на пяти различных машинах с процессорами Эльбрус:

1. Эльбрус 101-РС,
2. Эльбрус 401-РС,
3. Эльбрус-4.4,
4. Эльбрус 801-РС,
5. Эльбрус-8.4.

Основные характеристики тестируемых устройств приведены в таблице 1.

Была проведена экспериментальная оценка рассмотренных методов повышения быстродействия, а именно, проведены замеры среднего времени распознавания одного изображения, содержащего произвольным образом повернутый паспорт РФ, до и после использования данных методов. Это время не включало время загрузки изображения и файлов конфигурации из памяти. Усреднение выполнялось по набору из 800 изображений. Результаты приведены в таблице 2. Можно видеть, что оптимизация позволила повысить быстродействие системы в 5.3 раза, причем распараллеливание дало ускорение в 2.4 раза, а использование интринсиков — еще в 2.2 раза. Несмотря на то, что Эльбрус-401РС содержит 4 вычислительных ядра, не все части вычислительной системы были распараллелены, поэтому результирующее ускорение не достигло 4. Интринсики также существенно помогли повысить быстродействие, однако далеко не все операции были ускорены с их помощью. Тем не менее, было получено ускорение в 2.2 раза.

Таблица 3. Среднее время распознавания различных типов документов

Клиентский режим					
	Эльбрус 101-РС	Эльбрус 401-РС	Эльбрус-4.4	Эльбрус 801-РС	Эльбрус-8.4
Паспорт РФ	3.87 с	1.90 с	1.80 с	1.21 с	1.09 с
Биометрический паспорт РФ	3.33 с	1.85 с	1.80 с	1.10 с	1.05 с
ВУ РФ	4.24 с	2.12 с	1.81 с	1.24 с	1.09 с
ВУ	2.26 с	1.08 с	1.03 с	0.69 с	0.66 с
Великобритании					
Немецкие идентификационные карты	2.32 с	1.22 с	1.13 с	0.77 с	0.72 с
Листок нетрудоспособности	7.59 с	3.40 с	2.65 с	1.97 с	1.49 с
Серверный режим					
Паспорт РФ	—	1.27 с	0.36 с	0.43 с	0.11 с
Биометрический паспорт РФ	—	1.13 с	0.36 с	0.42 с	0.11 с
ВУ РФ	—	1.79 с	0.47 с	0.64 с	0.16 с
ВУ Великобритании	—	0.93 с	0.26 с	0.32 с	0.08 с
Немецкие идентификационные карты	—	0.99 с	0.26 с	0.37 с	0.10 с
Листок нетрудоспособности	—	2.22 с	0.66 с	0.86 с	0.22 с

Для оценки производительности всей распознающей системы использовалось среднее время распознавания одного изображения, содержащего произвольным образом повернутый документ заранее известного типа. Это время не включало время загрузки изображения и файлов конфигурации из памяти. Для документа каждого типа усреднение выполнялось по набору данных из 800 изображений.

В данной работе были рассмотрены 6 различных типов документов:

- паспорт РФ;
- биометрический паспорт РФ;
- водительское удостоверение (ВУ) РФ;
- водительское удостоверение (ВУ) Великобритании;
- немецкие идентификационные карты;
- листок нетрудоспособности.

В таблице 3 приведено среднее время распознавания документов каждого типа в режимах клиентского и серверного распознавания. Время распознавания на одноядерной машине Эльбрус-101РС составило 2.3 – 7.6 с для разных документов, поскольку они значительно различаются по объему распознаваемой информации. На остальных тестируемых устройствах время распознавания всех документов, кроме листка нетрудоспособности, не превышает 2 с в клиентском режиме и 1.8 с в серверном режиме. Можно видеть, что в клиентском режиме распознавание больше, чем в

4 потока, не дает значительного прироста производительности на всех документах, кроме листка нетрудоспособности. Этот результат связан со свойствами документов: в паспорте распознается 12 текстовых полей, а в водительском удостоверении и идентификационных картах – 7. В таких условиях распараллеленной оказывается не такая большая часть алгоритма распознавания. Листок нетрудоспособности содержит значительно больше полей, поэтому ускорение между 401-РС и Эльбрус-4.4 и 801-РС и Эльбрус-8.4 заметнее. Также стоит отметить, что 101-РС с одним ядром работает всего вдвое медленнее, чем 401 РС, поскольку в рассмотренном 101-РС стоит процессор Эльбрус-1С+ с обновленной микроархитектурой, поддерживающий исполнение до 25 операций за такт и работающий на частоте 950 МГц, а в рассмотренном 401-РС – Эльбрус-4С предыдущей версии.

В серверном режиме запускается несколько независимых процессов распознавания документов. Каждый вызов распознавания распараллелен так же, как и в предыдущем эксперименте, однако здесь время обработки включало время загрузки изображения из файла. Такой режим обеспечивает полную загрузку ядер процессора и позволяет более реалистично оценить производительность соответствующих устройств.

Результаты экспериментов показали, что серверные модули на основе процессоров Эльбрус демонстрируют ускорение в 3–4 раза за счет параллелизма на уровне задач. При этом сервер

Эльбрус-4.4 на 20–30% мощнее рабочей станции Эльбрус 801-РС. В свою очередь 801-РС практически в 3 раза быстрее своего предшественника 401-РС за счет повышения тактовой частоты и значительного усовершенствования архитектуры. Для Эльбрус-4.4 и Эльбрус 8.4 это соотношение сохранилось.

6. ЗАКЛЮЧЕНИЕ

Результаты проведенных экспериментальных исследований подтвердили работоспособность системы распознавания Smart IDReader на широком наборе вычислительных устройств с процессорами Эльбрус. Было показано, что процессор нового поколения Эльбрус-8С практически в 3 раза превышает по производительности Эльбрус-4С применительно к рассмотренной системе распознавания. Вычислительная мощность серверов на основе процессоров Эльбрус также практически в 4 раза превышает мощность рабочих станций за счет наличия большего числа процессорных ядер.

В работе было использовано основное преимущество VLIW-архитектур, — высокая степень параллелизма за счет широких командных слов, — наличие SIMD-расширения и параллелизм на уровне задач. В результате удалось добиться снижения времени распознавания в 5 раз, благодаря чему стало возможно применять систему Smart IDReader на практике, что и было показано на примере задач распознавания различных документов.

СПИСОК ЛИТЕРАТУРЫ

1. *Mastov A., Konovalenko I., Grigoryev A.* Application of random ferns for non-planar object detection, Proc. SPIE 9875, Eighth International Conference on Machine Vision, 98750M (December 8, 2015). 2015.
2. *Kopenkov V.N., Myasnikov V.V.* Detection and tracking of vehicles based on the video-registration information, Posters Proceedings of 23-rd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG 2015), Czech Republic, Plzen, June 8–12. 2015. P. 65–69.
3. *Арлазаров В.В.* Cognitive Form: Распределенная система потокового распознавания стандартных форм документов // Труды конференции “Распознавание образов и анализ изображений: новые информационные технологии” (РОАИ-6-2002). Великий Новгород, РФ, 21–26 октября. 2002. С. 41–45.
4. *Арлазаров В., Постников В., Шоломов Д.* Cognitive Forms система массового ввода структурированных документов // Управление информационными потоками, Сборник трудов Института системного анализа РАН. М., УРСС, 2002. С. 35–46.
5. *Omondi A.R., Rajapakse J.C.* FPGA Implementations of Neural Networks. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
6. *Farabet C., Poulet C., Han J.Y., LeCun Y.* CNP: An FPGA-based processor for Convolutional Networks. 2009. P. 32–37.
7. *Yang Q., Wang C., Gao Y., Qu H., Chang E.Y.* Inertial Sensors Aided Image Alignment and Stitching for Panorama on Mobile Phones, Proceedings of the 1st International Workshop on Mobile Location-based Service, MLBS’11, ACM, New York, NY, USA, 2011. P. 21–30.
8. *Lugman M.M., Gomez-Kramer P., and Ogier J.-M.* Mobile Phone Camera-Based Video Scanning of Paper Documents, Camera-Based Document Analysis and Recognition: 5th International Workshop, CBDAR 2013, Washington, DC, USA, August 23, 2013, Revised Selected Papers, M. Iwamura and F. Shafait, eds. 2014. P. 164–178.
9. *Ramisa A., Tapus A., De Mantaras R.L., Toledo R.* Mobile robot localization using panoramic vision and combinations of feature region detectors, IEEE International Conference on Robotics and Automation 2008: ICRA 2008: 19–23 May 2008, Pasadena, California, 2008. P. 538–543.
10. *Ким А.К., Бычков И.Н. и др.* Российские технологии “Эльбрус” для персональных компьютеров, серверов и суперкомпьютеров // Современные информационные технологии и ИТ-образование, М.: Фонд содействия развитию интернет-медиа, ИТ-образования, человеческого потенциала “Лига интернет-медиа”, 2014. № 10. С. 39–50.
11. *Ким А.К., Перекатов В.И., Ермаков С.Г.* Микропроцессоры и вычислительные комплексы семейства “Эльбрус”. СПб.: Питер, 2013. 272 с.
12. *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, М.: Научно-производственное объединение “Алмаз” им. акад. А.А. Расплетина. 2015. Т. 8. № 4. С. 64–68.
13. *Bezmaternykh P., Nikolaev D., Arlazarov V.L.* Textual Blocks Rectification Method Based on Fast Hough Transform Analysis in Identity Documents Recognition, The 10th International Conference on Machine Vision (ICMV 2017), November 13–15, Vienna, Austria, 2017.
14. *Stahlberg F., Vogel S.* The QCRI recognition system for handwritten arabic, International Conference on Image Analysis and Processing. 2015. P. 276–286.
15. *Чернов Т.С., Разумный Н.П., Кожаринов А.С., Арлазаров В.В.* Оценка качества входных изображений в системах распознавания видеопотока // Информационные технологии и вычислительные системы. 2017. № 4. С. 71–82.
16. *Usilin S., Nikolaev D., Postnikov V., Schaefer G.* Visual appearance based document image classification // International Conference on Image Processing, ICIP. 2010. P. 2133–2136.
17. *Бочаров Н.А., Лимонова Е.Е., Парамонов Б.Н., Усилин С.* Оптимизация для вычислительной архитектуры Эльбрус модифицированного метода Виола и Джонса // Труды ИСА РАН. 2017. Т. 67. № 4. С. 10–21.

18. *Chen N., Blostein D.* A survey of document image classification: problem statement, classifier architecture and performance evaluation // *International Journal of Document Analysis and Recognition (IJ DAR)*. 2007. V. 10. Issue 1. P. 1–16.
19. *Воронцов К.В.* Аддитивная регуляризация тематических моделей коллекций текстовых документов // *Доклады РАН*. 2014. Т. 456. № 3. С. 268–271.
20. *Junga K., Kimb K.I., Jain A.K.* Text information extraction in images and video: a survey // *Pattern Recognition*. 2004. P. 977–997.
21. *Слугин Д.Г., Арлазаров В.В.* Поиск текстовых полей документа с помощью методов обработки изображений // *Труды Института системного анализа РАН*. 2017. Т. 67. № 4. С. 65–73.
22. *Limonova E., Ilin D., Nikolaev D.P.* Improving Neural Network Performance on SIMD Architectures // *Eighth International Conference on Machine Vision (ICMV 2015)*, publ. code 98750L. 2015. V. 9875. P. 1–6.
23. *Limonova E., Sheshkus A., Nikolaev D.* Computational optimization of convolutional neural networks using separated filters architecture // *International Journal of Applied Engineering Research*. 2016. V. 11. № 11. P. 7491–7494.
24. *Лимонова Е., Шешкус А., Николаев Д.П., Иванова А., Ильин Д., Арлазаров В.Л.* Оптимизация быстродействия первых слоев глубоких сверточных нейронных сетей // *Вестник РФФИ*. 2016. № 4. С. 84–96.
25. *Limonova E., Sheshkus A., Ivanova A., Nikolaev D.P.* Convolutional neural network structure transformations for complexity reduction and speed improvement // *Pattern Recognition and Image Analysis*. 2018. № 1. P. 24–33.
26. *Ilin D., Limonova E., Arlazarov V.V., Nikolaev D.* Fast Integer Approximations In Convolutional Neural Networks Using Layer-By-Layer Training, *Proc. SPIE 10341*, Ninth International Conference on Machine Vision (ICMV 2016), publ. code 103410Q. 2017. P. 1–5.
27. *Гавриков М.Б., Пестрякова Н.В., Славин О.А., Фарсобица В.В.* Развитие метода полиномиальной регрессии и его практическое применение в задачах распознавания // *Препринты ИПМ им. М.В. Келдыша*. 2006. № 25. С. 1–22.
28. *Krizhevsky A., Sutskever I., Hinton G.E.* Imagenet classification with deep convolutional neural networks // *Advances in Neural Information Processing Systems*, 2012.
29. *Manzhikov T.V., Slavin O.A., Faradjev I.A., Janiszewski I.M.* Algorithms for Correcting Recognition Results Using N-grams. *ISSN Pattern Recognition and Image Analysis*. 2017. V. 4. № 27. P. 1054–6618. <https://doi.org/10.1134/S1054661817040125>
30. *Bulatov K., Manzhikov T., Slavin O., Faradjev I., Janiszewski I.* Trigram-based algorithms for OCR result correction. 2017. V. 10341. P. 103410O. <https://doi.org/doi.org/10.1117/12.2268559>
31. *Llobet R., Navarro-Cerdan J.R., Perez-Cortez J.-C., Arlandis J.* OCR Post-processing using Weighted Finite State Transducers, *Proceedings of the 12-th International Conf. of Frontiers in Handwriting Recognition, Kolkata (India)*. 2010. P. 2021–2024.
32. *Булатов К.Б.* Выбор оптимальной стратегии комбинирования кадровых результатов распознавания символа в видеопотоке // *Журнал “Информационные технологии и вычислительные системы”*. 2017. № 3. С. 45–55.
33. *Fiscus J.G.* A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER), *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*. 1997. P. 347–354.
34. *Bulatov K., Arlazarov V., Chernov T., Slavin O., Nikolaev D.* Smart IDReader: Document Recognition in Video Stream // *The 14th IAPR International Conference on Document Analysis and Recognition (ICDAR 2017)*, *CBDAR 2017: November 9–12, Kyoto, Japan, 2017*. P. 39–44.
35. *Арлазаров В.В., Арлазаров В.Л., Булатов К., Николаев Д.П., Полевой Д., Славин О.* Система распознавания документов в видеопоследовательности // *Государственный реестр полезных моделей Российской Федерации*.
36. Система программирования для платформ Эльбрус и МЦСТ-R, http://www.mcst.ru/sistema_programmirovaniya
37. Intel® Threading Building Blocks (Intel TBB), <https://www.threadingbuildingblocks.org/>