

УДК 004.8 528.7

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА КОМПЬЮТЕРНОГО ЗРЕНИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ ДЛЯ МОНИТОРИНГА ТЕХНОЛОГИЧЕСКИХ ОБЪЕКТОВ ПРЕДПРИЯТИЙ НЕФТЕГАЗОВОЙ ОТРАСЛИ

Зоев Иван Владимирович¹,
ivz3@tpu.ru

Марков Николай Григорьевич¹,
markovng@tpu.ru

Рыжова Светлана Евгеньевна¹,
ser1@tpu.ru

¹ Национальный исследовательский Томский политехнический университет,
Россия, 634050, г. Томск, пр. Ленина, 30.

Актуальность исследования обусловлена необходимостью создания современных компьютерных систем для мониторинга опасных технологических объектов предприятий нефтегазовой отрасли.

Цель: создание интеллектуальной системы компьютерного зрения беспилотных летательных аппаратов, позволяющей вести мониторинг опасных технологических объектов и анализ данных мониторинга в режиме реального времени на борту беспилотных летательных аппаратов.

Объекты: концепция построения интеллектуальной системы компьютерного зрения; новые архитектуры свёрточных нейронных сетей, аппаратно-реализованные на программируемых логических интегральных схемах; метод унификации вычислительных блоков и способы параллельных вычислений в аппаратных свёрточных нейронных сетях; алгоритмы помехоустойчивого кодирования/декодирования данных при обменах сообщениями между наземной и бортовой компонентами интеллектуальной системы компьютерного зрения.

Методы: методы классификации и детектирования объектов на изображениях с помощью свёрточных нейронных сетей; методы глубокого обучения свёрточных нейронных сетей; методы проектирования программно-аппаратных систем.

Результаты. Проведён анализ современного состояния исследований в области систем мониторинга опасных технологических объектов предприятий нефтегазовой отрасли; разработана концепция создания интеллектуальной системы компьютерного зрения на основе беспилотных летательных аппаратов для мониторинга опасных объектов. Базовой в концепции является идея анализа изображений, полученных при мониторинге технологических объектов и прилегающих к ним территорий, непосредственно на борту беспилотных летательных аппаратов в режиме реального времени. Более того, показано, что для обеспечения такого анализа в реальном времени необходимо применять аппаратно-реализованные свёрточные нейронные сети. Для интеллектуальной системы компьютерного зрения разработаны архитектуры свёрточных нейронных сетей из перспективных подклассов LeNet5 и YOLO; предложены алгоритмы помехоустойчивого кодирования/декодирования данных при обмене сообщениями между наземной и бортовой компонентами системы компьютерного зрения; разработан оригинальный метод организации вычислений в аппаратных свёрточных нейронных сетях на программируемых логических интегральных схемах, отличающийся от известных использованием унифицированных вычислительных блоков; предложены новые способы параллельных вычислений в слоях таких свёрточных нейронных сетей. Разработана архитектура вычислительного устройства беспилотных летательных аппаратов, включающего блоки аппаратной свёрточной нейронной сети и кодер/декодер данных. Устройство создано на основе системы на кристалле Cyclone V SX компании Altera; получены первые результаты исследования эффективности этого устройства; разработано программное обеспечение наземной компоненты системы компьютерного зрения.

Ключевые слова:

Беспилотные летательные аппараты, мониторинг опасных технологических объектов нефтегазовой отрасли, система компьютерного зрения, свёрточные нейронные сети, программируемые логические интегральные схемы.

Введение

В настоящее время в России и во многих зарубежных промышленно развитых странах существует проблема мониторинга и выявления противоправных действий на опасных промышленных объектах. Кроме неё остаётся нерешённой проблема своевременного выявления возгораний на таких объектах и на занимаемых ими территориях. Решение этих проблем, в первую очередь, актуально для предприятий нефтегазовой отрасли, имеющих опасные технологические объекты на промыслах при добыче и подготовке углеводородного сырья (УВС) и протяжённые (сотни и тысячи километров) магистральные нефте- и газопроводы для транспортировки УВС. Для решения таких про-

блем все чаще начинают применять космические аппараты (КА) и беспилотные летательные аппараты (БПЛА) с установленной на них специальной аппаратурой для мониторинга этих технологических объектов.

В данной статье предлагается оснастить каждый БПЛА для мониторинга интеллектуальной системой компьютерного зрения (СКЗ). Такая СКЗ включает видеокамеру, тепловизор и вычислительное устройство, позволяющее прямо на борту беспилотника решать задачи распознавания объектов различной физической природы (автомобили, технологические объекты, люди, спецтехника и т. п.) и очаги возгорания на видеоснимках и снимках с тепловизора, сделанных на территории

мониторинга. В основу интеллектуальной СКЗ положены свёрточные нейронные сети (СНС), причём используется их аппаратная реализация. Приводится разработанная архитектура вычислительного устройства, устанавливаемого на БПЛА, и результаты исследования его эффективности.

Задачи мониторинга технологических объектов предприятий нефтегазовой отрасли

Для ряда регионов РФ характерна большая протяженность (сотни и тысячи километров) магистральных нефте- и газопроводов (линейных технологических объектов), обслуживаемых нефте- и газотранспортными предприятиями. Более того, в случае газовой отрасли можно говорить о наличии большого числа газораспределительных станций и весьма развитой сети газопроводов среднего и низкого давления. У нефтегазодобывающих предприятий кроме опасных технологических объектов в виде установок подготовки УВС на промыслах также имеются межпромысловые газопроводы, конденсатопроводы и нефтепроводы. Для всех предприятий нефтегазовой отрасли актуально решение следующих классов задач:

- мониторинг и выявление противоправных посягательств (нарушение охранных зон и зон минимальных безопасных расстояний, несанкционированный отбор УВС и т. п.) и террористических угроз для технологических объектов, в первую очередь для трубопроводов;
- мониторинг и выявление пожаров и технологических загрязнений почвы с целью обеспечения пожарной безопасности и охраны окружающей среды на технологических объектах для добычи, транспорта и хранения УВС и прилегающих к ним территорий.

По мнению многих специалистов и учёных такой мониторинг опасных технологических объектов и их охранных зон следует осуществлять с помощью КА и (или) БПЛА. При этом съёмка объектов и охранных зон в зависимости от решаемых задач должна выполняться в оптическом, инфракрасном, а иногда и в миллиметровом диапазонах длин волн [1]. Учитывая, что доступные аппараты из летающих сегодня на орбите КА с датчиками высокого разрешения позволяют вести съёмку территории большинства регионов России не чаще, чем один раз в сутки, и тот факт, что значительную часть года над этой территорией присутствует облачность, предпочтение следует отдать съёмке территорий, содержащих технологические объекты, с помощью БПЛА. Это подчёркивает актуальность и практическую значимость работ по созданию интеллектуальной СКЗ на основе БПЛА для мониторинга путём съёмки технологических объектов и их охранных зон. Также актуальной является проблема раннего выявления очагов возгорания на таких объектах и на прилегающих к ним территориях.

Проанализируем современное состояние исследований в области систем мониторинга вышеука-

занных объектов. В последние годы ОАО «Газпром космические системы», являющееся дочерним предприятием ПАО «Газпром», интенсивно реализует проект по разработке системы дистанционного зондирования Земли «СМОТР». Эта система позволит вести аэрокосмический мониторинг технологических объектов и территорий предприятий ПАО «Газпром» с использованием собственных КА «Ямал-300» и «Ямал-400» и БПЛА в оптическом, инфракрасном и миллиметровом диапазонах длин волн [1]. Предполагается, что кроме мониторинга и выявления противоправных посягательств на технологические объекты система «СМОТР» будет выявлять очаги возгораний вблизи таких объектов. Отметим, что все изображения (снимки) с КА и БПЛА поступают в наземный сегмент системы – центр сбора и обработки данных, где снимки обрабатываются и анализируются (интерпретируются) по классическим алгоритмам распознавания объектов на изображениях. Отсутствие анализа снимков в реальном времени на борту КА и БПЛА не позволит своевременно принять решение и выслать тревожную группу к месту инцидента и, в итоге, пресечь посягательство на технологический объект.

За рубежом также имеются системы слежения на основе БПЛА за территориями с опасными технологическими объектами. Снимки получают с помощью видеокамер высокого разрешения, установленных на БПЛА, и передают их в наземные пункты, где они анализируются, чаще всего операторами. Анализ проводится с некоторым запаздыванием относительно времени несанкционированного доступа на объект. Имеются попытки автоматизировать процесс анализа снимков, при этом в качестве алгоритмов распознавания объектов на изображениях в основном используются традиционные методы классификации объектов. Однако окончательное решение по инциденту и в этих случаях обычно принимает оператор.

В нашей стране и за рубежом продолжает оставаться острой проблема раннего выявления пожаров. Для её решения за рубежом во многих странах сооружается сеть вышек, оборудованных сложной аппаратурой обнаружения дыма, видеокамерами высокого разрешения и т. п. [2]. Однако охватить большие территории с технологическими объектами, особенно находясь среди мощных лесных массивов, такими сетями не удаётся. Поэтому за рубежом и в России для выявления пожаров используются КА, например, зарубежные LEO и GEO, Landsat5, Terra, Aqua и т. п. Для детектирования огня слабой интенсивности на них устанавливаются радиометры высокого разрешения AVHRR, делают снимки земной поверхности с помощью спектрорадиометров MODIS и т. д. [2]. В ряде случаев анализ по сложным алгоритмам этих данных даёт положительные результаты [3]. Наблюдается тенденция к выполнению анализа данных в автоматическом режиме по ряду алгоритмов, включая нейросетевые, на борту КА [2]. К сожалению, данные с

КА для значительной части территории многих стран обновляются только раз в двое суток и только для малого числа стран – раз в сутки. Это не позволяет решить проблему выявления пожаров на ранней стадии, что еще раз подчеркивает актуальность использования БПЛА для мониторинга территорий с целью выявления очагов возгорания.

В настоящее время в России мониторинг территорий для выявления на ранних стадиях очагов возгораний ведется эпизодически с помощью БПЛА самолётного типа, при этом продолжительность полёта аппарата составляет 2,5–4 часа. Зарубежные аппараты такого типа находятся в воздухе обычно 4–5 часов. Для локальной разведки (удаление на 2–3 км) используются БПЛА вертолётного типа. На каждом БПЛА обычно устанавливается видеокамера и тепловизор (датчики ИК-диапазона) [4]. Передача полученных при облёте территории видеоизображений и термоизображений осуществляется в режиме реального времени в наземный пункт службы лесоохраны (или заинтересованного предприятия, или соответствующей службы МЧС). Такой режим передачи данных с БПЛА накладывает весьма жёсткие требования на ширину полосы пропускания радиоканала и на приёмо-передающие устройства БПЛА. Обработка и анализ изображений на борту БПЛА не ведутся. Находящийся в наземном пункте оператор осуществляет просмотр видеоизображений, ведёт наблюдение за появлением термоточек на снимках с тепловизора и принимает решение о наличии очага возгорания. Отметим, что при интерпретации снимков оператор опирается на свой опыт и интуицию. Его решение об очаге возгорания является субъективным и может оказаться ошибочным.

В последние годы как в России, так и за рубежом интенсивно ведутся исследования по автоматизации в наземных пунктах процессов распознавания очагов возгорания на изображениях, полученных с КА и БПЛА. Некоторые результаты этих исследований, основанные на использовании классических методов и алгоритмов распознавания объектов на изображениях при дистанционном зондировании Земли, позволили уменьшить субъективный фактор при принятии решений оператором [2, 5]. Однако уровень автоматизации и точность распознавания объектов на изображениях пока невысоки. Ряд исследователей предлагают реализовать автоматический режим анализа изображений с КА и БПЛА в наземных пунктах и выявлять очаги возгорания без участия оператора. Они считают, что для этого необходимо использовать искусственные нейронные сети, обученные с помощью технологий глубокого обучения [6]. В этом направлении получены пока только первые результаты [7].

Концепция создания интеллектуальной системы компьютерного зрения

В отличие от описанных выше вариантов использования БПЛА для мониторинга технологиче-

ских объектов и выявления противоправных посягательств на них, а также для мониторинга территорий с целью выявления очагов возгорания нами предлагается идея анализа полученных при мониторинге изображений непосредственно на борту БПЛА в режиме реального времени. Для этого на основе БПЛА необходимо создать интеллектуальную бортовую компоненту СКЗ. При анализе изображений такая компонента на борту БПЛА должна автоматически распознавать (классифицировать или детектировать) объекты различной физической природы. По мнению многих исследователей весьма эффективными для реализации таких задач распознавания объектов являются искусственные нейронные сети. Среди таких нейросетей предпочтение будем отдавать классу СНС, показывающих по сравнению с нейросетями других классов наиболее высокую точность распознавания объектов на изображениях [6, 8, 9]. Более того, для обеспечения анализа изображений в СКЗ в масштабе реального времени предлагается опираться на перспективную идею аппаратной реализации СНС.

Кратко проанализируем современное состояние исследований в области архитектуры и обучения СНС как наиболее предпочтительного класса нейросетей для решения поставленных задач анализа данных мониторинга. Архитектура любой СНС предполагает наличие свёрточных слоёв (англ. convolutional layers), слоёв подвыборки (англ. pooling layers) и функции активации [10]. Исходное для анализа изображение, полученное в результате съёмки, подаётся на вход первого свёрточного слоя. Базовой для каждого свёрточного слоя является процедура свёртки. Результат операции свёртки – это изображение, называемое картой признаков.

Процедура подвыборки (англ. pooling) является базовой для слоёв подвыборки и уменьшает размерность каждой карты признаков. Входная карта признаков при реализации этой процедуры разбивается на окна обычно размером 2×2 элемента. В случае алгоритма подвыборки с использованием максимальных значений элементов (англ. max pooling) в выходное изображение записывается элемент с максимальным значением из каждого окна. В соответствии с результатами исследований из [11], этот алгоритм производительнее других, что важно в нашем случае, поскольку можно увеличить производительность разрабатываемой СКЗ.

В качестве функции активации всё чаще используется функция ReLU (Rectified Linear Unit) [12]. ReLU заменяет все отрицательные элементы на нулевые. Её использование имеет по сравнению с другими функциями активации ряд преимуществ, весьма важных для создания СКЗ.

Отметим, что архитектура каждой нейросети из класса СНС определяется числом свёрточных и подвыборочных слоёв, а также параметрами архитектуры свёрточных слоёв. Сегодня некоторые исследователи, учитывая специфику решаемых ими

прикладных задач, уже конструируют СНС с архитектурой, отвечающей их требованиям [13]. При этом обязательны исследования эффективности таких архитектур с целью поиска компромисса между скоростью работы нейросети и точностью распознавания с её помощью объектов на изображениях. Особенно такой компромисс важен в нашем случае при создании бортовой компоненты СКЗ для работы в реальном времени.

В 1998 г. Ян ЛеКун с коллегами предложил новую нейросеть, получившую название СНС. Архитектура такой сети называется LeNet5 и сегодня считается классической [6]. Более того, сейчас существует целый подкласс СНС с архитектурами, полученными с учётом вышеуказанных особенностей свёрточных нейросетей и подобных LeNet5 [6, 10, 13]. В последние годы появились другие подклассы СНС с более сложными архитектурами, в первую очередь, подкласс СНС большой размерности AlexNet [9, 13, 14]. Среди СНС современных архитектур весьма перспективными являются AlexNet, ResNet и YOLO [9, 10, 14]. По мнению ряда исследователей, сегодня именно YOLO имеет наилучшую скорость вычислений [13, 14]. Это важно при создании СКЗ реального времени для БПЛА. Однако задача оценки точности распознавания объектов на изображениях для интересующей нас предметной области с помощью СНС YOLO требует исследований. Анализ показывает, что для аппаратной реализации перспективны СНС подкласса LeNet5 и подкласса YOLO.

Для извлечения ключевых признаков весовые коэффициенты процедуры свёртки настраиваются с использованием обучающей выборки. Заметим, что класс СНС относится к глубоким нейронным сетям, для которых естественным является применение при обучении технологии глубокого обучения (англ. deep learning) [6, 14]. Существует довольно много обучающих выборок, собранных для решения различного рода прикладных задач [15]. Обучение СНС проводится методом обратного распространения ошибки и с использованием одного из методов (алгоритмов) оптимизации [16].

Важным положением при создании интеллектуальной СКЗ является принцип разделения такой СКЗ на две компоненты: бортовую и наземную. При этом интеллектуальной, в первую очередь, является бортовая компонента СКЗ, основанная на использовании для анализа изображений перспективных СНС подклассов LeNet5 и YOLO, подвергнутых глубокому обучению. Такие СНС должны автоматически распознавать (классифицировать или детектировать) на изображениях объекты различной физической природы (автомобили, людей и т. д.). Разрабатываемые новые архитектуры СНС этих подклассов будут отличаться от известных архитектур по ряду архитектурных признаков, в первую очередь, числом свёрточных слоев.

Особенностью развиваемого подхода к созданию интеллектуальной СКЗ является то, что бортовая компонента СКЗ должна функционировать в

режиме реального времени. Иными словами, анализ изображений и принятие решений об инцидентах на территории мониторинга должны осуществляться в темпе поступления данных с видеокamеры и тепловизора СКЗ. Темп поступления данных определяется скоростью полета БПЛА и характеристиками видеокamеры и тепловизора (разрешающая способность, угол обзора и т. д.). С учетом высоких требований со стороны СНС к производительности вычислительного устройства бортовой компоненты СКЗ из-за необходимости проведения вычислений в режиме реального времени предлагается аппаратная реализация новых архитектур СНС на программируемых логических интегральных схемах (ПЛИС), имеющих минимальное энергопотребление и небольшой вес.

Наземная компонента СКЗ располагается в наземном пункте соответствующей службы мониторинга территорий и в качестве ядра должна содержать геоинформационную систему (ГИС) для работы оператора с пространственными данными, например, при прокладке маршрутов БПЛА. На электронную карту автоматически с помощью этой системы должны также наноситься данные о местонахождении инцидентов, выявленных с помощью бортовой компоненты СКЗ. Основным принципом создания такой ГИС – как можно больше использовать функционал наиболее известной свободно-распространяемой ГИС.

Учитывая высокую вероятность возникновения ошибок при передаче по радиоканалу команд с наземного пункта в БПЛА и при передаче данных об инцидентах с БПЛА в наземную компоненту СКЗ, важным положением концепции является необходимость использовать помехоустойчивое кодирование таких данных. Для этого следует разработать алгоритмы кодирования/декодирования и исследовать возможность создания на их основе кодера-декодера (кодека) на ПЛИС в составе бортовой компоненты СКЗ и программно-реализованного декодера для наземной компоненты СКЗ.

С учетом приведенных подходов и принципов создания интеллектуальной СКЗ разработаны основные требования, которым должны удовлетворять наземная и бортовая компоненты такой СКЗ. Число требований превышает 100.

Новые архитектуры свёрточных нейронных сетей для системы компьютерного зрения

Рассмотрим разработанные для СКЗ новые архитектуры СНС перспективных подклассов LeNet5 и YOLO. В предложенной нами для решения задачи классификации объектов на изображениях архитектуре СНС (рис. 1), подобной классической архитектуре LeNet5, используются 3 свёрточных слоя, 2 слоя подвыборки, а в качестве функции активации после каждого слоя свёртки применяется оператор ReLU [17]. СНС может классифицировать принадлежность объекта к одному из 10 классов. Назначение класса объекту по результатам работы СНС осуществляется с помощью известной

процедуры Softmax [14]. Архитектурные параметры каждого из слоёв показаны на рис. 1.

В большинстве СНС конечные слои являются полносвязными, в случае нашей СНС это также имеет место.

Из рис. 1 видно, что после первого свёрточного слоя, результатом которого является 6 карт признаков размером 26×26 элементов, идёт слой подвыборки с окном 2×2 элемента и шагом 2. Для каждого такого окна выполняется процедура подвыборки, которая проводится по алгоритму выбора максимальных значений элементов. После этого слоя получается 6 карт признаков размером 13×13 элементов. Другой слой подвыборки с таким же окном 2×2 элемента и шагом 2 выполняется после второго слоя свёртки. На его выходе имеют 32 карты признаков размером 5×5 элементов.

Предложенная архитектура СНС отличается от классической архитектуры LeNet5 наличием трёх свёрточных слоёв вместо двух и наличием только одного полносвязного слоя вместо двух. Имеются также отличия в параметрах свёрточных слоёв. Все это ведёт к увеличению количества карт признаков и нацелено на увеличение точности классификации объекта на изображении. Сравнение предложенной архитектуры с другими известными СНС подкласса LeNet5 позволяет считать её по ряду архитектурных признаков оригинальной.

Для решения задачи создания вычислительного устройства бортовой компоненты СКЗ необходимо не только обучить программно реализованную СНС, но и использовать полученные весовые коэффициенты свёрточных слоёв в дальнейшем при аппаратной реализации этой нейросети на ПЛИС.

Нами предложена методика переноса весовых коэффициентов для их использования в аппаратной реализации СНС [18]. Отметим, что для СНС предложенной архитектуры проведены исследования её эффективности, в том числе её аппаратной реализации на ПЛИС, по различным критериям. Исследования показали её высокую эффективность при решении задачи классификации объектов на изображениях.

Перспективным среди архитектур нейросетей подкласса YOLO для решения задач детектирования объектов на изображениях считаются СНС архитектуры YOLOv2 (имеет 22 свёрточных слоя) и более простая архитектура tiny-YOLO, имеющая только 9 свёрточных слоёв [19]. Точность детектирования объектов на изображениях с помощью СНС YOLOv2 по методике mAP составляет 76,8 % на выборке Pascal VOC 2007. В архитектуре нейросети tiny-YOLO не используется добавление карт признаков из предыдущих слоёв в последующие, как это делается в СНС YOLOv2. Однако точность детектирования и скорость выполнения этой нейросети довольно близки к соответствующим значениям в случае использования архитектуры YOLOv2. В этой связи перспективной для последующей аппаратной реализации на ПЛИС является СНС tiny-YOLO.

В статье [20] приведены результаты исследования, которое показало, что увеличение количества слоёв СНС не ведёт к повышению точности детектирования объектов на изображениях. Вместе с этим, была предложена новая архитектура с использованием ResNet-модулей, где данные на входе слоя подаются также и на вход следующего за ним через два слоя. Это позволяет использовать меньшее количество слоёв сети при более высокой точности детектирования объектов. Возможна комбинация модулей, когда в Inception-модуль встраивается ResNet-модуль. На рис. 2 приведена детализированная схема Inception-ResNet-модуля. Такой подход и различные комбинации модулей представлены в статье [20]. В результате сокращается объём вычислений в сравнении с более глубокими нейросетями, но сохраняется точность детектирования объектов. Используем данный подход для развития архитектуры СНС tiny-YOLO, заменив часть слоёв на Inception-ResNet-модули в различных комбинациях. Так как выход Inception-ResNet-модуля имеет размер в 256 карт признаков и объединяется со входом, то изменим количество карт признаков четвёртого свёрточного слоя СНС tiny-YOLO со 128 на 256 карт признаков. Также

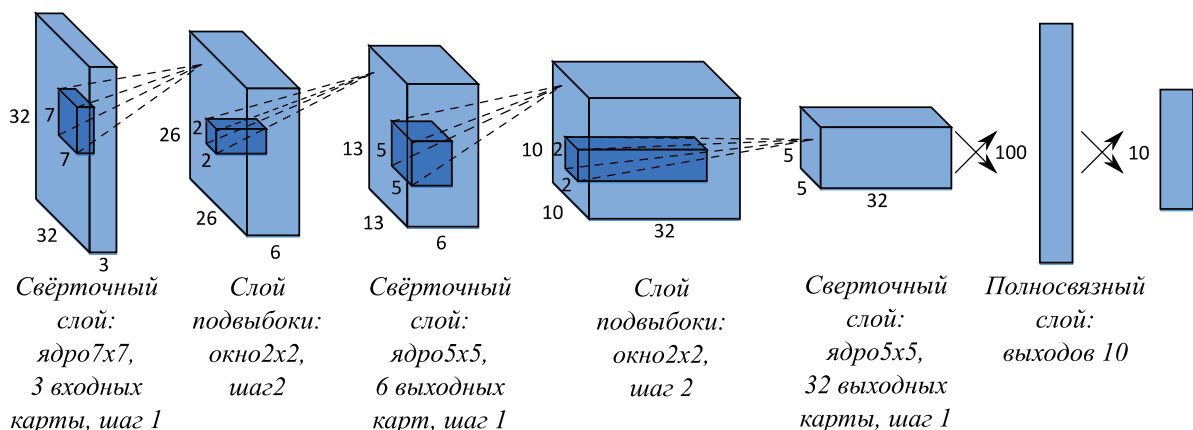


Рис. 1. Предложенная архитектура СНС из подкласса LeNet5

Fig. 1. CNN's proposed architecture from subclass LeNet5

уменьшим количество карт признаков в последующих слоях свёртки этой нейросети.

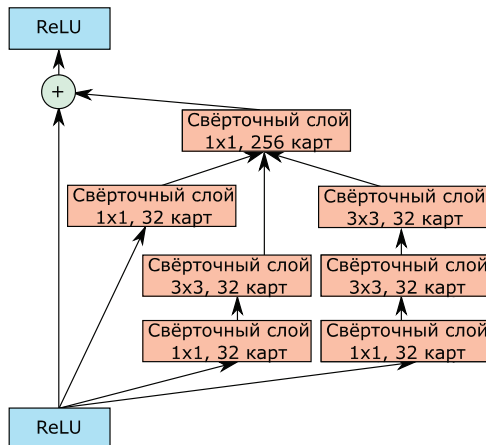


Рис. 2. Inception-ResNet-модуль

Fig. 2. Inception-ResNet-module

В табл. 1 представлена новая архитектура СНС tiny-YOLO-Inception-ResNet.

Таблица 1. Архитектура СНС tiny-YOLO-Inception-ResNet

Table 1. tiny-YOLO-Inception-ResNet CNN architecture

№ слоя Layer no.	Тип слоя Type of layer	Число ядер свёртки Number of convolutional kernels	Размеры ядра/шаг Kernel size /stride	Размеры входного изображения/карт признаков Input images size/features maps
1	Свёрточный Convolutional	16	3×3/1	416×416×3
2	Подвыборки Pooling	–	2×2/2	416×416×16
3	Свёрточный Convolutional	32	3×3/1	208×208×16
4	Подвыборки Pooling	–	2×2/2	208×208×32
5	Свёрточный Convolutional	64	3×3/1	104×104×32
6	Подвыборки Pooling	–	2×2/2	104×104×64
7	Свёрточный Convolutional	256	3×3/1	52×52×64
8	Подвыборки Pooling	–	2×2/2	52×52×256
9	Inception-ResNet	–	–	26×26×256
10	Свёрточный Convolutional	128	3×3/1	26×26×256
11	Подвыборки Pooling	–	2×2/2	26×26×128
12	Свёрточный Convolutional	256	3×3/1	13×13×128
13	Подвыборки Pooling	–	2×2/1	13×13×256
14	Свёрточный Convolutional	128	3×3/1	13×13×256
15	Свёрточный Convolutional	125	1×1/1	13×13×128
16	Детектор Detector	–	–	–

Предложим еще несколько новых архитектур СНС на базе нейросети tiny-YOLO. Так, используем последовательно по два и три Inception-ResNet-модуля, которые следуют друг за другом, и получим соответственно архитектуры tiny-YOLO-Inception-ResNet2 и tiny-YOLO-Inception-ResNet3. Также параллельно используем два и три таких же модуля, имеющих один вход и конкатенацию их выходных карт признаков, что даёт новые архитектуры tiny-YOLO-Inception-ResNet-x2 и tiny-YOLO-Inception-ResNet-x3.

Результаты наших исследований каждой новой архитектуры СНС изложены в [21]. Исследования для этих архитектур проводились по скорости выполнения, сложности последующей аппаратной реализации и точности детектирования объектов на изображениях из тестовой выборки PascalVOC 2007 [22]. В этой выборке объекты на изображениях относятся к 20 классам. Скорость выполнения каждой из архитектур определялась как среднее время выполнения СНС при детектировании объектов одного изображения из тестовой выборки, состоящей из 4952 изображений.

Выяснилось, что при разработке интеллектуальной СКЗ следует обратить внимание на архитектуры tiny-YOLO-Inception-ResNet2 и tiny-YOLO-Inception-ResNet3, поскольку именно для них имеет место баланс между точностью детектирования объектов и скоростью выполнения этих СНС. Однако по критерию минимального времени на анализ одного изображения и по требуемым вычислительным ресурсам при аппаратной реализации СНС предпочтение следует отдать нейросети архитектуры tiny-YOLO-Inception-ResNet2. СНС такой архитектуры была выбрана нами для реализации в вычислительном устройстве БПЛА.

Алгоритмы помехоустойчивого кодирования/декодирования данных и их реализация в системе компьютерного зрения

Обмен сообщениями между бортовой и наземной компонентами СКЗ ведётся по радиоканалу. При передаче данных по такому каналу могут возникать ошибки двух типов: независимые ошибки или пакеты ошибок. Обычно используются коды либо обнаруживающие, либо обнаруживающие и исправляющие ошибки. Среди кодов, обнаруживающих ошибки, наиболее универсальным является циклический код CRC (англ. Cyclic redundancy code – циклический код контроля) [23]. Он построен на вычислении контрольной суммы, которая является функцией передаваемого сообщения. Вычисленное значение добавляется в конец передаваемых данных. Однако в условиях повышенной вероятности возникновения ошибок при передаче данных с БПЛА в наземную компоненту СКЗ кодов, обнаруживающих ошибки, может быть недостаточно. Необходимы коды, способные оперативно исправлять пакетные ошибки.

Поскольку в случае инцидента на территории мониторинга с БПЛА будут передаваться данные о классе объекта и его координатах и два видеоснимка

ка (в случае выявления очага возгорания снимок с тепловизора и снимок этого же участка с видеокамеры), имеющие большой объём, и эти данные следует кодировать, то существует необходимость снижения нагрузки на канал передачи данных. Для решения этой проблемы предлагается оригинальный способ двухэтапного наложения избыточности: для обнаружения ошибок на первом этапе и дополнительно на втором этапе для исправления ошибок. Суть его в следующем. Изображения и данные об инциденте перед передачей делятся на блоки. Для каждого из блоков рассчитывается контрольная сумма CRC, которая добавляется в конец блока (рис. 3, а). Эта сумма позволит обнаружить ошибки в данных, пришедших в наземную компоненту СКЗ. В случае необходимости исправления ошибок декодер посылает на БПЛА сообщение о проведении усиленного кодирования этих блоков. Тогда на БПЛА к данным, закодированным по схеме, представленной на рис. 3, а, вторым этапом будет добавлена избыточность в виде пакета, необходимая для исправления ошибок в кодовой комбинации (рис. 3, б). Декодер БПЛА предназначен для декодирования команд, посылаемых с наземной компоненты СКЗ.

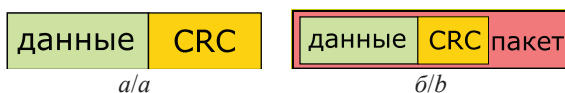


Рис. 3. Схема кодирования данных (а) и схема усиленного кодирования данных (б)

Fig. 3. Data encoding scheme (a) and enhanced data encoding scheme (b)

Рассмотрим подход к построению кодера наземной компоненты СКЗ. Длина кодируемых команд, передаваемых на БПЛА, составляет до 2-х байт. К ним так же, как в кодере бортовой компоненты, будет рассчитываться и добавляться контрольная сумма CRC4 (рис. 3, а). Затем сразу же будет добавляться избыточная часть, необходимая в последующем для обнаружения и исправления ошибок (рис. 3, б). Для реализации наземной компоненты СКЗ разработан оригинальный двухэтапный способ декодирования блоков данных с БПЛА. На первом этапе декодер, реализующий этот способ, для каждого блока вычисляет контрольную сумму CRC. Если она верна, то декодируются следующие блоки данных. В случае несовпадения контрольных сумм декодер посылает команду на БПЛА о необходимости усиленного кодирования. На втором этапе ведётся декодирование принятых с БПЛА блоков в соответствии со схемой на рис. 3, б с избыточностью с целью обнаружения и исправления ошибок.

Обычно вычисление контрольной суммы CRC ведётся по матричному или табличному алгоритмам [24]. Анализ показывает, что для любого из этих алгоритмов требуются достаточно скромные ресурсы ПЛИС при аппаратной реализации контрольной суммы CRC.

Для исправления ошибок использовался полиномиальный циклический код. Кодирование данных осуществлялось по алгоритму систематического кодирования, когда кодовая комбинация формируется в виде двух отдельных блоков: информационного блока – длиной m бит и контрольного блока – длиной k бит.

Начало

Шаг 1. Информационная часть сообщения расширяется нулями до значения $x^{(n-1)}$; $n=m+k$;

Шаг 2. Полученная кодовая комбинация делится на образующий полином $g(x)$. Получаем остаток от деления $R(x)$;

Шаг 3. Полученный остаток суммируется с многочленом, сформированным на *Шаге 1*, по модулю 2. Результатом суммирования является закодированная информационная последовательность.

Конец

Такой алгоритм является наиболее распространенным в системах передачи данных и простым в своей реализации. Для декодирования таких кодов нами применялся циклический метод. Основная его идея заключается в итерационном процессе вычисления синдромов ошибки (вычисления остатка от деления кодового слова с ошибкой на образующий полином) и определения его веса [25]. Этот метод применяется для кодов, которые исправляют независимые ошибки. Использовались такие две модификации циклического метода декодирования, которые позволяют применять его для кодов, исправляющих пакетные ошибки.

Аппаратно-программная реализация вычислительного устройства БПЛА

Каждая из рассмотренных выше новых архитектур СНС при реализации на ПЛИС будет иметь вычислительные блоки двух типов, реализующих процедуры свёртки и подвыборки, соответственно, в слоях свёртки и подвыборки СНС. С учетом того, что в СНС выходная карта признаков каждого слоя формируется независимо от других слоев и СНС при проведении вычислений в каждом слое присутствует естественный параллелизм, вычислительные блоки соответствующего типа могут выполняться параллельно внутри слоя. Из-за использования в предложенных архитектурах СНС процедур свёртки и подвыборки с разными значениями параметров число одновременно выполняемых в слоях вычислительных блоков будет отличаться и будет зависеть от конкретного слоя.

При реализации на ПЛИС архитектуры СНС подкласса LeNet5 (рис. 1) для выполнения первого слоя свёртки назначим 6 вычислительных блоков свёртки (по числу выходных карт признаков слоя), для второго слоя – 6 вычислительных блоков подвыборки, для реализации третьего свёрточного слоя – 32 блока свёртки и т. д. По сути, этот новый способ (далее – Способ 1) организации вычислений в аппаратной СНС позволяет реализовать одноэтапную конвейерную архитектуру СНС на ПЛИС, а в каждом из слоёв соответствующий вычисли-

тельный блок формирует свою выходную карту признаков параллельно с другими блоками слоя, вычисляющими каждый свою карту признаков. Однако в этом конвейере время на реализацию вычислений для каждого слоя различно, что ведет к простоям ряда вычислительных блоков. Наши оценки показывают, что время на вычисления в первом слое значительно превосходит время на вычисления в других слоях. Это указывает на необходимость разработки новых, сокращающих время простоя блоков, способов и методов организации вычислений при аппаратной реализации СНС на ПЛИС.

Нами предложено при организации вычислений в аппаратных СНС использовать вычислительные возможности не только ПЛИС, но и других компонентов современных систем на кристалле (СнК). Так, в выбранной нами системе на кристалле Cyclone V SX компании Altera кроме ПЛИС имеется двухъядерный процессор ARM Cortex A9, который имеет прямой доступ к внешней памяти. Архитектура рассматриваемой СнК позволяет организовать прямой доступ аппаратно реализованной СНС к этой внешней памяти. Реализация такого способа взаимодействия ПЛИС с внешней памятью позволит, в свою очередь, выполнять некоторые операции, отличные от процедур свёртки и подвыборки, не на ПЛИС, а на самом процессоре СнК.

Опираясь на этот способ, предложен оригинальный метод организации вычислений в аппаратной СНС на ПЛИС. Он отличается от известных методов использованием унифицированных вычислительных блоков свёртки и подвыборки. Унификация блоков свёртки/подвыборки достигается путем извлечения параметров блоков, обычно задающихся на этапе их синтеза, и размещения их в отдельную изменяемую область памяти ПЛИС, называемую конфигурационной областью. Это позволит использовать блоки соответствующего типа в слоях СНС, имеющих разные архитектурные параметры. Реализация метода позволяет считать, что число задействованных в аппаратной СНС вычислительных блоков может быть переменным и определяется только ресурсами ПЛИС. Масштабирование по числу блоков ведётся как в целом для СНС, так и для её отдельных слоёв. Это позволит значительно сократить используемые вычислительные ресурсы ПЛИС и реализовать различные архитектуры СНС без реконфигурации ПЛИС.

При проведении унификации кроме реализации конфигурационной области памяти также необходимо организовать доступ к этой области памяти процессору СнК и создать дополнительное программное обеспечение, в первую очередь, в виде многофункционального драйвера аппаратной СНС.

Учитывая, что каждый элемент выходной карты признаков вычисляется независимо от других, и опираясь на метод унифицированных вычислительных блоков, кроме Способа 1 предложены другие новые способы организации параллельных

вычислений в СНС на ПЛИС. Суть второго из них, названного Способ 2, в том, что параллельно в слое свёртки выполняются вычислительные блоки, каждый из которых рассчитывает свой элемент выходной карты признаков. Тогда число таких блоков определяется как произведение размеров выходной карты признаков, умноженное на глубину L_i – количество таких карт i -го слоя. В условиях ограниченных ресурсов ПЛИС (в том числе при ограничении на число вычислительных блоков) этот способ может быть модифицирован следующим образом. Один блок будет последовательно использоваться при расчёте двух или более соседних элементов или даже всех элементов строки выходной карты признаков. Для дальнейшего сокращения числа вычислительных блоков возможна другая модификация Способа 2, когда каждый блок используется для расчёта элементов четвертой части или даже половины всех строк одной выходной карты признаков.

Способ 3 заключается в том, что несколько блоков в слое параллельно вычисляют один элемент выходной карты признаков. Наконец, можно предложить ещё ряд способов организации вычислений, комбинируя Способ 2 и Способ 3 по слоям СНС.

На рис. 4 приведена архитектура вычислительного устройства БПЛА, созданная на СнК Cyclone V SX компании Altera.

Вычислительное устройство состоит из двух блоков внешней памяти типа DDR3 и СнК. В свою очередь, СнК включает процессорную систему и ПЛИС. Процессорная система состоит из двух процессорных ядер ARM Cortex A9, контроллера внешней памяти DDR3 и контроллера прерываний. СнК также имеет каналы, позволяющие реализовать два физических подключения к ней блоков внешней памяти. Первый из каналов связывает ПЛИС с первым из блоков внешней памяти, а второй канал – процессорную систему с другим блоком. Использование второго канала позволило организовать взаимодействие между ПЛИС и процессорной системой через контроллер DDR3 последней.

Связи между ПЛИС и процессорной системой организованы по принципу ведущий–ведомый. Так, для связи F2H между ПЛИС и процессорной системой ПЛИС выступает в роли ведущего устройства, а процессорная система – в роли ведомого. При использовании канала H2F процессорная система выступает в роли ведущего устройства, а ПЛИС – в роли ведомого. Имеется прямое подключение по каналу F2H SDRAM ПЛИС к контроллеру памяти DDR3 процессорной системы, при этом ПЛИС выступает в роли ведущего устройства, а контроллер – в роли ведомого. Также имеется шина прерываний IRQ, которая соединяет ПЛИС с контроллером прерываний процессорной системы.

В ПЛИС разработаны и размещены: оригинальный контроллер прямого доступа к памяти – кон-

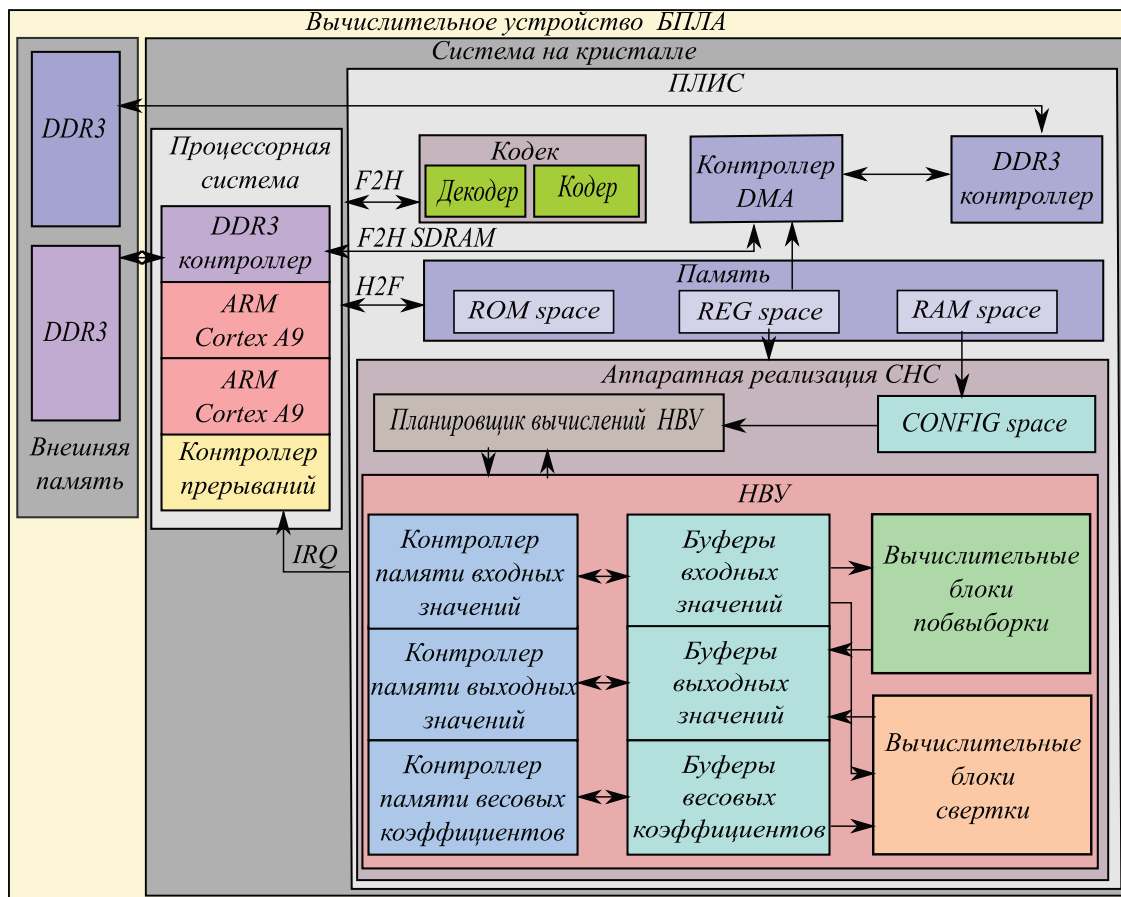


Рис. 4. Архитектура вычислительного устройства БПЛА

Fig. 4. Architecture of computing device for unmanned aerial vehicles (UAVs)

трон DMA (англ. Direct memory access– DMA), собственный контроллер внешней памяти (DDR3 контроллер), внутренняя память, кодек (кодер и декодер помехоустойчивых кодов) и блок «Аппаратная реализация СНС». Внутренняя память ПЛИС включает в себя блок данных об устройстве (назван ROM space), блок конфигурации СНС (RAM space) и блок управления нейровычислительным устройством (НВУ), названный REG space. В памяти CONFIG space содержится конфигурация слоя СНС, исполняемого в текущий момент.

Контроллер DMA соединен с контроллером DDR3 ПЛИС и с контроллером памяти DDR3 процессорной системы, что обеспечивает не только прямой доступ к обоим блокам внешней памяти DDR3, но и позволяет осуществлять взаимодействие между блоками памяти процессорной системы и ПЛИС.

Внутренняя память RAM space и REG space, включенная в ПЛИС, конфигурируется процессорной системой по каналу H2F. Чтение данных из внутренней памяти ROM space также происходит по этому каналу.

Кодек (кодер-декодер) соединен с процессорной системой каналом F2H.

Блок «Аппаратной реализации СНС» является оригинальным и включает в себя НВУ, планировщик вычислений в НВУ и память CONFIG space. В свою очередь, НВУ имеет буферы для входных и выходных значений (хранятся, соответственно, исходное изображение и выходная карта признаков вычисляемого слоя), а также буферы весовых коэффициентов СНС. Для каждого типа буфера имеется соответствующий контроллер памяти (контроллер входных значений, контроллер выходных значений и контроллер весовых коэффициентов). НВУ имеет набор унифицированных вычислительных блоков свертки и унифицированных вычислительных блоков подвыборки.

Для алгоритма вычисления контрольной суммы CRC и алгоритма систематического кодирования разработан кодер на ПЛИС в составе вычислительного устройства БПЛА. Его функциональная схема приведена на рис. 5. Кодер работает в двух возможных режимах в соответствии с оригинальным способом двухэтапного наложения избыточности. Первый режим работы кодера вычисляет только контрольную сумму CRC для обнаружения ошибок. Вторым режимом является в случае получения команды с наземной компоненты СКЗ о необходимости усиленного кодирования.

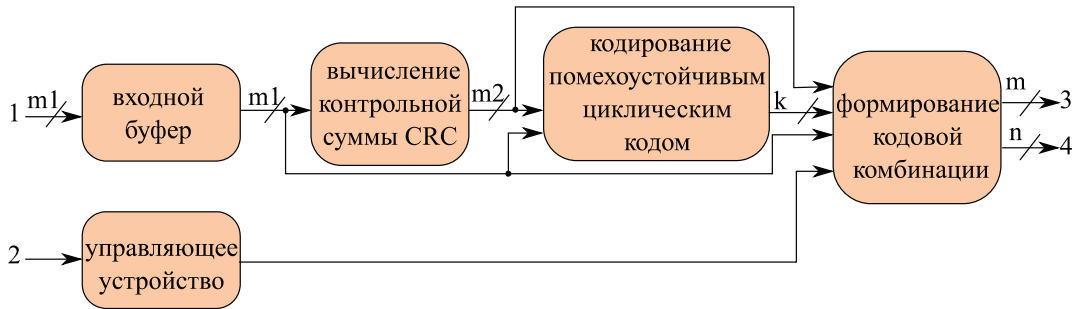


Рис. 5. Функциональная схема кодера в составе вычислительного устройства БПЛА: 1 – входная кодовая комбинация, 2 – сигнал для выбора режима кодирования (CRC или двухэтапное); $m1$ – длина блока кодируемых данных; $m2$ – разрядность рассчитанной для блока входных данных контрольной суммы CRC; k – разрядность избыточности, добавленной кодером помехоустойчивого кода; $m=m1+m2$ – длина блока данных с добавочной контрольной суммой CRC; $n=m+k$ – разрядность кодовой комбинации, закодированной кодом, исправляющим пакетные ошибки; 3 – кодовая комбинация с блоком обнаружения ошибки; 4 – кодовая комбинация с блоками обнаружения и исправления ошибки

Fig. 5. Encoder functional circuit of the UAV's computing device: 1 is the input code combination; 2 is the signal to select the encoding mode (CRC or two-step); $m1$ is the length of encoding data; $m2$ is the bit length of the CRC checksum; k is the bit length of the redundancy added by the error-correcting code encoder; $m=m1+m2$ – the length of the data block with added CRC checksum; $n=m+k$ – the length of the code combination encoded by the error-correcting code, which corrects burst errors; 3 is the code combination with error detection block; 4 is the code combination with error detection and correction blocks

Также на ПЛИС была разработана функциональная схема блока декодирования (рис. 6) в составе вычислительного устройства БПЛА.

Поступающие с наземной компоненты СКЗ данные декодируются в два этапа. На первом этапе обнаруживается и исправляется пакетная ошибка, а на втором этапе путём вычисления контрольной суммы CRC контролируется корректность работы декодера на первом этапе.

Данная архитектура разработана на основе бюджетной СнК Cyclone V SX компании Altera, однако может применяться для более производительных СнК.

Вычислительное устройство создано на основе отладочной платы Terasic SoCkit с СнК Cyclone V SX. Устройство имеет описанную выше архитектуру (рис. 4) и реализует СНС подкласса LeNet5 и подкласса YOLO.

Основные программные модули в вычислительном устройстве БПЛА, реализуемые для двухъядерного процессора Arm Cortex A9:

- модуль извлечения параметров архитектуры СНС из файла protobuf описания архитектуры СНС;
- модуль извлечения данных весовых коэффициентов из файла весовых коэффициентов архитектуры СНС;
- модуль принятия решений об инциденте на территории мониторинга;
- модуль формирования данных и отправки их на наземный пункт;
- модуль взаимодействия с аппаратной реализацией СНС;
- модуль приема команд с пункта управления БПЛА;
- модуль операционной системы на базе ядра Linux (Драйвер);
- модуль поиска аномальных термоточек на снимках с тепловизора.

Отметим, что до начала работы вычислительного устройства БПЛА необходимо описать в виде файла формата protobuf архитектуру СНС, кото-

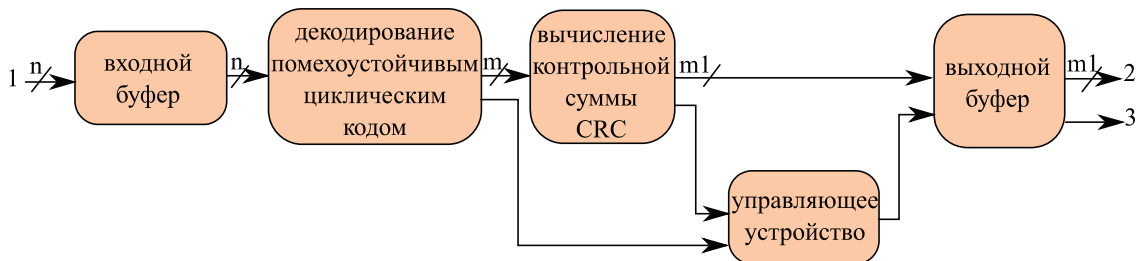


Рис. 6. Функциональная схема декодера в составе вычислительного устройства БПЛА: 1 – кодовая комбинация, содержащая блоки обнаружения и исправления ошибок; n – длина кодовой комбинации; m – разрядность блока, содержащего данные и контрольную сумму CRC; $m1$ – разрядность декодированной части сообщения, 2 – информация, полученная после декодирования принятой комбинации; 3 – сигнал об обнаружении неисправимой ошибки в принятой комбинации

Fig. 6. Decoder functional circuit of the UAV's computing device: 1 is the code combination, which includes error detection and correction clocks; n is the length of code combination; m is the bit length of block, containing the data and CRC checksum; $m1$ is the bit length of decoded part of the data; 2 is the data obtained after decoding of receiving information; 3 is the signal about detection of unrecoverable error in the received code combination

рая будет исполняться на ПЛИС. Также необходимо сформировать файл весовых коэффициентов СНС. Затем программный модуль «Извлечение параметров архитектуры СНС» извлекает из первого файла описание архитектуры СНС и передаёт их Драйверу. Последний после приёма значений параметров СНС производит разметку адресного пространства внешней памяти. После этого каждый из программных модулей в нужный момент времени выполняет возложенные на него функции.

Исследование эффективности вычислительного устройства беспилотных летательных аппаратов

Исследовалось устройство, реализующее предложенную нами архитектуру СНС подкласса LeNet5 (рис. 1), позволяющую решать задачу классификации объектов на изображениях. Для этого сначала обучали программно реализованную СНС этой архитектуры. Существует большое число обучающих выборок, собранных для решения различного рода задач. С учётом поставленной задачи классификации объектов на изображении хорошо подходит большая база изображений MNIST [26]. Она содержит 60000 обучающих изображений и 10000 тестовых пар (изображение-метка). Обучение СНС проводилось с использованием библиотеки Caffe [27], реализующей метод обратного распространения ошибки [16]. Точность классификации объектов для программно реализованной СНС этой архитектуры составила при использовании формата 32 разрядных чисел с плавающей запятой 98,7 % и принята за эталонную точность. Полученные при обучении весовые коэффициенты были перенесены в вычислительное устройство по нашей методике, изложенной в [18].

Для сокращения необходимых вычислительных ресурсов ПЛИС можно уменьшить разрядность форматов чисел с плавающей запятой. Поэтому были проведены эксперименты по исследованию точности классификации объектов на изображениях в зависимости от разрядности формата чисел с плавающей запятой. В экспериментах участвовали три варианта вычислительного устройства, у которых на ПЛИС были реализованы базовые операции над числами в формате с плавающей запятой различной разрядности. В табл. 2 приведены результаты этих экспериментов, для сравнения второй строкой указан результат по точности классификации объектов с помощью программно реализованной СНС той же архитектуры (эталонная точность). Видим, что результаты по точности классификации с помощью устройства при разрядности чисел 16 и 14 бит близки к значению эталонной точности классификации объектов. Это означает, что возможна экономия ресурсов ПЛИС путём разумного снижения разрядности чисел с плавающей запятой.

Другой эксперимент был нацелен на исследование производительности вычислительного устройства, в котором используется 16 разрядный формат чисел с плавающей запятой. Для этого прово-

дились теоретические расчёты с применением инструментария ModelSim компании MentorGraphics [28] и эксперименты по оценке времени классификации каждого изображения из 10000 тестовых изображений выборки MNIST с помощью устройства. Исследовалась производительность вычислительного устройства в двух вариантах. В первом варианте устройства используется распараллеливание вычислений в слоях по описанному выше Способу 1 (вариант обозначен С1). Второй вариант устройства – это модификация первого варианта, при которой в первом и втором слоях СНС использовалось распараллеливание вычислений по Способу 2 (вариант устройства обозначен С2). Во втором варианте в каждом из этих двух слоёв выходную карту признаков рассчитывают 2 вычислительных блока. В теоретических расчётах принято, что каждый вычислительный блок работает на частоте 50 МГц. Результаты, полученные теоретическим и экспериментальным путём, приведены в табл. 3.

Таблица 2. Зависимость точности классификации объектов от разрядности чисел с плавающей запятой

Table 2. Dependence of objects classification accuracy on the digit capacity of floating-point numbers

Аппаратная и программная реализация СНС Hardware and software implementation of CNN	Точность распознавания, % Accuracy of recognition, %			
	12 бит bits	14 бит bits	16 бит bits	32 бит bits
Вычислительное устройство БПЛА UAV computing device	97,03	98,51	98,66	–
Программная реализация СНС Software implementation of CNN	–	–	–	98,7

Таблица 3. Производительность при выполнении аппаратной СНС для двух вариантов вычислительного устройства

Table 3. Performance for hardware-based CNN for two variants of computing device

№ слоя Layer no.	Тип слоя Layer type	Кол-во операций в слое Number of operations in a layer	Кол-во вычислительных блоков Number of computing units		Время работы аппаратной СНС для классификации одного изображения, мс Run time of hardware-based CNN for one image recognition, ms			
			C1	C2	Теоретическое Theoretical		Экспериментальное Experimental	
					C1	C2	C1	C2
1	Свёрточный Convolutional	596232	6	12	1,987	0,994	4,310	3,495
2	Подвыборки Pooling	4056	6	12	0,014	0,007	0,391	0,244
3	Свёрточный Convolutional	388800	32	32	0,243	0,243	0,726	0,733
4	Подвыборки Pooling	3200	32	32	0,002	0,002	0,274	0,221
5	Свёрточный Convolutional	80000	100	100	0,016	0,016	1,281	1,278
6	Свёрточный Convolutional	1000	10	10	0,002	0,002	0,221	0,214
Итого/Summary					2,264	1,270	7,202	6,085

Видим, что теоретически рассчитанное время, затрачиваемое на классификацию одного изображения, довольно сильно отличается от полученного экспериментальным путём для каждого их вариантов вычислительного устройства. Это обусловлено значительными затратами времени на операции чтения данных из внешней памяти в ПЛИС. По этой же причине увеличение производительности вычислений в первых двух слоях СНС при удвоении числа вычислительных блоков в каждом из этих слоёв не привело к её двукратному росту, который ожидался исходя из теоретически рассчитанных результатов.

Исходя из первых полученных результатов исследования производительности вычислительного устройства, можно сделать вывод о том, что необходимы дальнейшие, в том числе комплексные, исследования, направленные на изучение эффективности различных предложенных способов распараллеливания вычислений в аппаратной СНС. Также следует указать на проблему избыточного числа обращений аппаратной СНС к внешней памяти, решение которой должно привести к росту производительности вычислительного устройства.

Выполнена также оценка эффективности декодера на ПЛИС в составе вычислительного устройства БПЛА. Эффективность оценивалась для информационного блока $m=16$ бит и различных размеров пакета ошибок (p изменяется от 5 до 8 бит). В табл. 4 приведены значения характеристик декодера для кодов с различными параметрами.

Таблица 4. Характеристики декодера для кодов с различными параметрами

Table 4. Decoder characteristic for codes with different parameters

n	m	p	Логические ячейки, шт Logic cells, numbers	Время работы декодера, нс Decoder run time, ns
30	16	5	1981	15,5
31	16	6	2286	17,7
31	16	7	2238	15,7
42	16	8	4378	20,5

Из табл. 4 следует, что декодер для кода с параметрами (31, 16, 7) наиболее приемлем по количеству требующихся логических ячеек ПЛИС и по быстродействию. Кодер на ПЛИС в отличие от декодера требует ещё меньше логических ячеек ПЛИС.

Программное обеспечение наземной компоненты системы компьютерного зрения

В соответствии с концепцией создания интеллектуальной СКЗ её наземная компонента располагается в пункте управления БПЛА соответствующей службы мониторинга территорий и должна иметь специализированное программное обеспечение (ПО). Одним из требований этой концепции является наличие в составе ПО ядра, которое представляет из себя программные средства свободно распространяемой геоинформационной си-

стемы (ГИС). В качестве ГИС-ядра при выполнении проекта была выбрана QGIS версии 2.8 [29]. QGIS – это свободно-распространяемая кроссплатформенная ГИС, состоящая из настольной компоненты QGIS Desktop, серверной компоненты QGIS Server и компоненты QGIS Web Client для веб-доступа к настольной компоненте. Сегодня эта ГИС адаптирована к требованиям наземной компоненты СКЗ и имеет набор функций, позволяющий оператору наземного пункта выполнять базовые операции над пространственными данными и в итоге управлять БПЛА. С помощью инструментального модуля для осуществления разработки программ, входящего в QGIS, могут быть созданы и интегрированы в ПО этой ГИС новые программные модули.

В настоящее время решена задача кодирования/декодирования данных в наземной компоненте СКЗ. При этом предложен подход к построению кодера, в соответствии с которым рассчитывается и добавляется контрольная сумма CRC и сразу же добавляется избыточная часть (рис. 3, б). Для реализации декодирования разработан оригинальный двухэтапный способ декодирования блоков данных, принятых с БПЛА. На основе этого подхода к кодированию и двухэтапного способа декодирования данных разработаны соответствующие алгоритмы. При этом учтено, что ПО, реализующее эти алгоритмы, будет функционировать в условиях отсутствия жёсткого ограничения на использование вычислительных ресурсов наземной компоненты СКЗ. Эти алгоритмы были реализованы на языке C# в виде веб-сервиса кодирования и веб-сервиса декодирования данных. Веб-сервисы интегрированы с программными средствами ГИС QGIS.

Заключение

Предложена идея анализа изображений, полученных при мониторинге опасных технологических объектов и прилегающих к ним территорий, непосредственно на борту БПЛА в режиме реального времени. Для её реализации разработана концепция интеллектуальной СКЗ, базирующаяся на использовании аппаратных СНС для детектирования или классификации объектов различной физической природы в масштабе реального времени. Для аппаратной реализации на ПЛИС в качестве перспективных выбраны СНС подклассов LeNet5 и YOLO и разработаны новые архитектуры СНС этих подклассов.

Разработан оригинальный метод организации вычислений, учитывающий унификацию вычислительных блоков, в аппаратных СНС на ПЛИС, и новые способы параллельных вычислений в слоях СНС с новыми архитектурами. Создано вычислительное устройство БПЛА на основе системы на кристалле Cyclone V SX. Особенностью архитектуры устройства является наличие кроме реализованной на ПЛИС аппаратной СНС блоков кодера и декодера для помехоустойчивого кодирования/декодирования данных в СКЗ. Разработано ПО бортовой и наземной компонентов СКЗ.

Получены первые обнадеживающие результаты исследований созданного вычислительного устройства БПЛА, исходя из которых был сделан вывод о необходимости дальнейших комплексных исследований эффективности устройства с целью минимизации требующихся вычислительных ресурсов ПЛИС.

СПИСОК ЛИТЕРАТУРЫ

1. Программа развития компании ОАО «Газпром космические системы». Система «СМОТР». URL: <http://www.gazprom-spacesystems.ru/new-pro/> (дата обращения: 11.01.2019).
2. Ahmad A.A. Alkhatib A review on Forest Fire Detection Techniques // International Journal of Distributed Sensor Networks. – 2014. – V. 10. – Iss. 3. – P. 1–12. URL: <https://journals.sagepub.com/doi/full/10.1155/2014/597368/> (дата обращения: 11.01.2019).
3. Analysis of the moderate resolution imaging spectroradiometer contextual algorithm for small fire detection / Wanting Wang, J.J. Qu., Xianjun HaO, Yongqiang Liu // Journal of Applied Remote Sensing. – 2009. – V. 3. – 031502.
4. Лесные пожары. Беспилотное воздушное судно ZALA 421-16ЕМ Концерна Калашникова. URL: <http://zala.aero/category/applications/monitoring-pozharov/> (дата обращения: 12.01.2019).
5. Построение системы работы с данными прибора MODIS для решения задач мониторинга лесных пожаров и их последствий / Е.В. Флитман, И.В. Балашов, М.А. Бурцев, А.А. Галеев, В.А. Егоров, Р.В. Котельников, Е.А. Лупян, А.А. Мазуров, А.М. Матвеев, А.А. Прошин // Современные проблемы дистанционного зондирования Земли из космоса. – 2011. – Т. 8. – № 1. – С. 127–138.
6. Николенко С., Кабури А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – СПб: ИД «Питер», 2017. – 328 с.
7. Колесенков А.Н., Костров Б.В., Ручкин В.Н. Нейронные сети мониторинга чрезвычайных ситуаций по данным ДЗЗ // Известия Тульского государственного университета. Технические науки. – 2014. – № 5. – С. 220–225.
8. Goyal S., Benjamin P. Object Recognition Using Deep Neural Networks: a survey. – 2014. URL: <https://arxiv.org/pdf/1412.3684.pdf> (дата обращения 08.01.2019).
9. Image net large scale visual recognition challenge / O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Kholsa, M. Bernstein, A.C. Berg, L. Fei-Fei // International Journal of Computer Vision. – 2015. – V. 15. – Iss. 3. – P. 211–252.
10. El-Sawy A., Hazem E.L.B., Loev M. CNN (convolutional neural networks) for hand-written Arabic digits recognition based on LeNet-5 // Proc. of the International Conference on Advanced Intelligent Systems and Informatics 2016. – 2016. – P. 566–575. – DOI: 10.1007/978-3-319-48308-5_54.
11. Scherer D., Miiller A., Behuke S. Evaluation of pooling operations in convolutional architectures for object recognition // Artificial Neural Networks – ICANN 2010. 20th International Conference. – Thessaloniki, Greece, 15–18 September, 2010. Springer-Verlag Berlin Heidelberg, 2010. – P. III. – P. 92–101.
12. Krizhevsky A., Sutskever I., Hinton G.E. Image net classification with deep CNN // Proc. of the 20th International Conference on Neural Information Processing Systems (ICONIP 2013). – Daejeon, Korea, November 3–7, 2013. – V. 1. – P. 1097–1105.
13. Танк Т. Итоги развития компьютерного зрения за 2016 год. URL: <https://habrahabr.ru/post/346140/> (дата обращения 15.04.2019).
14. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. – М.: ДМК Пресс, 2017. – 652 с.

Авторы благодарят к.т.н., доцента А.Н. Мальчукова за помощь в постановке задачи помехоустойчивого кодирования и декодирования данных и аспиранта А.П. Береснева за консультации по свёрточным нейронным сетям.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-47-700010_p_a.

15. Road and building detection datasets. URL: <http://www.cs.toronto.edu/~vmnih/data> (дата обращения: 15.04.2019).
16. Садовников П. Методы оптимизации нейронных сетей. URL: <https://habrahabr.ru/post/318970/> (дата обращения 10.01.2019).
17. Устройство на основе ПЛИС для распознавания рукописных цифр на изображениях / И.В. Зоев, А.П. Береснев, Н.Г. Марков, А.Н. Мальчуков // Компьютерная оптика. – 2017. – Т. 41. – № 6. – С. 938–949.
18. Береснев А.П., Зоев И.В., Мальчуков А.Н. Методика переноса весов нейронной сети из программной в аппаратную реализацию // Сб. трудов XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых. – Томск: Изд-во ТПУ, 2016. – Т. 1. – С. 22–23.
19. YOLO9000: Better, Faster, Stronger. URL: <https://arxiv.org/pdf/1612.08242.pdf> (дата обращения 15.04.2019).
20. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. – URL: <https://arxiv.org/abs/1602.07261> (дата обращения 14.05.2018).
21. Береснев А.П., Зоев И.В., Марков Н.Г. Исследование свёрточных нейронных сетей класса YOLO для мобильных систем детектирования объектов на изображениях // GraphiCon 2018. 28-я Международная конференция по компьютерной графике и машинному зрению. Труды конференции. – Томск, 2018. – С. 196–199.
22. Pascal VOC database. URL: <https://pjreddie.com/projects/pascal-voc-dataset-mirror/> (дата обращения 15.04.2019).
23. Morelos-Zaragoza R.H. The art of error correcting coding. 2nd ed. – Baffins Lane, Chichester, West Sussex, England: John Wiley & Sons, Ltd, 2002. – 269 p.
24. Мыцко Е.А., Мальчуков А.Н. Исследование программных реализаций алгоритмов вычисления CRC (cyclic redundancy code) совместных с PKZIP, WINRAR, ETHERNET // Известия Томского политехнического университета. – 2013 – Т. 322. – № 5. – С. 170–175.
25. Рыжова С.Е., Мыцко Е.А. Разработка адаптивной структуры быстродействующего декодера БЧХ-кода (Боуза–Чоудхури–Хоквингема-кода) и его аппаратная реализация на примере кодовой комбинации длиной в 15 бит на ПЛИС (на программируемой логической интегральной схеме) // Научная сессия ТУСУР-2017: материалы Международной научно-технической конференции студентов, аспирантов и молодых ученых, посвященной 55-летию ТУСУРа. – Томск, 2017. – Т. 5. – С. 148–151.
26. The MNIST database of handwritten digits. URL: <http://yann.lecun.com/exdb/mnist> (дата обращения 15.04.2019).
27. Caffe: Convolutional Architecture for Fast Feature Embedding / Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell // Proc. of the 22nd ACM Multimedia. – Orlando, Florida, USA, November 3–7, 2014. – P. 675–678.
28. Quartus II Handbook V. 3: Verification. URL: https://www.altera.com/content/dam/altera-ww/global/en_US/pdfs/literature/hb/qts/qts_qii5v3.pdf (дата обращения 15.05.2019).
29. QGIS is a free, open source, crossplatform GIS (geographic information system). URL: <https://github.com/qgis/QGIS> (дата обращения: 31.01.2019).

Поступила 12.06.2019 г.

Информация об авторах

Зоев И.В., аспирант отделения информационных технологий Инженерной школы информационных технологий и робототехники Национального исследовательского Томского политехнического университета.

Марков Н.Г., доктор технических наук, профессор отделения информационных технологий Инженерной школы информационных технологий и робототехники Национального исследовательского Томского политехнического университета.

Рыжова С.Е., аспирант отделения информационных технологий Инженерной школы информационных технологий и робототехники Национального исследовательского Томского политехнического университета.

UDC 004.8 528.7

INTELLIGENT COMPUTER VISION SYSTEM FOR UNMANNED AERIAL VEHICLES FOR MONITORING TECHNOLOGICAL OBJECTS OF OIL AND GAS INDUSTRY

Ivan V. Zoev¹,

ivz3@tpu.ru

Nikolay G. Markov¹,

markovng@tpu.ru

Svetlana E. Ryzhova¹,

ser1@tpu.ru

¹ National Research Tomsk Polytechnic University,
30, Lenin Avenue, Tomsk, 634050, Russia.

The relevance of the research is caused by the necessity to develop modern computer vision systems for monitoring hazardous technological objects of oil and gas industry.

The main aim of the research is to develop the intelligent computer vision system for unmanned aerial vehicles, which allows monitoring dangerous technological objects and analyzing the monitoring data in real-time on the board of the unmanned aerial vehicle.

Objects: the concept of construction of intelligent computer vision system; new architectures of convolutional neural networks hardware-based using field programmable gate array; the method of unification of computing blocks and ways of parallel calculation in hardware-based convolutional neural networks; algorithms of error-correction encoding and decoding data for exchanging message between ground and airborne components of the intelligent computer vision system.

Methods: methods of detection and classification objects in images using convolutional neural networks; convolutional neural network deep learning methods; methods of designing software and hardware systems.

Results. We have been analyzed the current state of research in the field of monitoring hazardous technological objects of the oil and gas industry and developed the concept of construction of intelligent computer vision system for unmanned aerial vehicles for monitoring dangerous objects. The idea of analyzing the images, obtained at monitoring of technological objects and surrounding areas, directly onboard of the unmanned aerial vehicle in real time was the base in this concept. Moreover, it is shown that the use of hardware-based convolutional neural networks for providing such analysis in real time is required. The authors developed the convolutional neural networks architectures for computer vision system from promising subclasses LeNet5 and YOLO and proposed the algorithms of error-correction data encoding/decoding for messages exchanging between these components, considering the specifics of ground and airborne components. The authors developed the original method of organizing calculation in hardware-based convolutional neural networks using field programmable gate array, which differs from the known ones by using the unified computing blocks and new ways of parallel calculation in layers in these convolutional neural networks. They proposed the architecture of computing device of the unmanned aerial vehicle which includes the blocks of the hardware-based convolutional neural networks and the data encoder/decoder. This device is based on the Altera Cyclone V SX system-on-a-chip. The paper demonstrates the first results of studying the device efficiency. The authors developed the software for the ground component of the computer vision system.

Key words:

Unmanned aerial vehicles, monitoring hazardous technological objects of oil and gas industry, computer vision system, convolutional neural networks, field programmable gate array.

The authors appreciate Cand. Sc., associate professor A.N. Malchukov for his support in data error-correction encoding and decoding task and post graduate student A.P. Beresnev for advising in convolutional neural networks.

The reported study was funded by RFBR, project no. 18–47–700010 p_a.

REFERENCES

1. *Programma razvitiya kompanii OAO «Gazprom kosmicheskie sistemy». Sistema «SMOTR»* [The development program of OJSC «Gazprom space systems». «SMOTR» system]. Available at: <http://www.gazprom-spacesystems.ru/new-pro/> (accessed 11 January 2019).
2. Ahmad A.A. Alkhatib. A review on Forest Fire Detection Techniques. *International Journal of Distributed Sensor Networks*, 2014, vol. 10, Iss. 3, pp. 1–12. Available at: <https://journals.sagepub.com/doi/full/10.1155/2014/597368/> (accessed 11 January 2019).
3. Wanting Wang, Qu J.J., Xianjun HaO, Yongqiang Liu. Analysis of the moderate resolution imaging spectroradiometer contextual algorithm for small fire detection. *Journal of Applied Remote Sensing*, 2009, vol. 3, 031502.
4. *Lesnye pozhary. Bepilotnoe vozdušnoe sudno ZALA 421–16EM Kontserna Kalashnikova* [Forest fires. Unmanned aircraft ZALA 421–16EM of Kalashnikov Concern]. Available at: <http://zala.aero/category/applications/monitoring-pozharov/> (accessed 12 January 2019).
5. Flitman E.V., Balashov I.V., Bourtsev M.A., Galeev A.A., Egorov V.A., Kotelnikov R.V., Loupian E.A., Mazurov A.A., Matveev A.M., Proshin A.A. Organization of the MODIS instrument data processing system for the tasks of monitoring forest fires and their aftereffects. *Current problems in remote sensing of the Earth from space*, 2011, vol. 8, no. 1, pp. 127–138. In Rus.
6. Nikolenko S., Kaburin A., Arkhangelskaya E. *Glubokoe obuchenie. Pogruzhenie v mir neyronnykh setey* [Deep learning. Immersion in the world of neural networks]. St-Petersburg, Piter Publ., 2017. 328 p.

7. Kolesenkov A.N., Kostrov B.V., Ruchkin V.N. Neural network monitoring of emergencies according ERS. *Proceedings of the Tomsk State University Technical science*, 2014, no. 5, pp. 220–225. In Rus.
8. Goyal S., Benjamin P. *Object Recognition Using Deep Neural Networks: a survey*, 2014. Available at: <https://arxiv.org/pdf/1412.3684.pdf> (accessed: 08 January 2019).
9. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A.C., Fei-Fei L. Image Net Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015, vol. 115, Iss. 3, pp. 211–252.
10. El-Sawy A., Hazem E.L.B., Loey M. CNN for hand-written Arabic digits recognition based on LeNet-5. *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*. 2016. pp. 566–575. DOI: 10.1007/978-3-319-48308-5_54.
11. Scherer D., Miiller A., Behuke S. Evaluation of pooling operations in convolutional architectures for object recognition. *Artificial Neural Networks – ICANN 2010. 20th International Conference*. Thessaloniki, Greece, 15–18 September 2010. Springer-Verlag Berlin Heidelberg, 2010. P. III, pp. 92–101.
12. Krizhevsky A., Sutskever I., Hinton G.E. Image Net classification with deep CNN. *Proceedings of the 20th International Conference on Neural Information Processing Systems (ICONIP 2013)*. Daegu, Korea, November 3–7, 2013. Vol. 1, pp. 1097–1105.
13. Tank T. *Itogi razvitiya kompyuternogo zreniya za 2016 god* [Results of the computer vision development for 2016]. Available at: <https://habrahabr.ru/post/346140/> (accessed 15 April 2019).
14. Gudfellow Y.A., Bengio I., Courville A. *Glubokoe obuchenie* [Deep learning]. Moscow, DMK Press, 2017. 652 p.
15. *Road and building detection datasets*. Available at: <http://www.cs.toronto.edu/~vmnih/data> (accessed 15 April 2019).
16. Sadovnikov P. *Metody optimizatsii neyronnykh setey* [Neural networks optimization methods]. Available at: <https://habrahabr.ru/post/318970/> (accessed 10 January 2019).
17. Zoev I.V., Beresnev A.P., Markov N.G., Malchukov A.N. FPGA based device for handwritten digits recognition on images. *Computer optics*, 2017, vol. 41, no. 6, pp. 938–949. In Rus.
18. Beresnev A.P., Zoev I.V., Malchukov A.N. Metodika perenosov vesoov neyronnoy seti iz programmnoy v apparatnuyu realizatsiyu [The method of transferring weights of the neural network from software implementation to hardware one]. *Sbornik trudov XIV Mezhdunarodnoy nauchno-prakticheskoy konferentsii studentov, aspirantov i molodykh uchennykh* [MSIT. Proc. of the XIV International scientific-practical conference of students, graduate students and young scientists]. Tomsk, TPU Publ., 2016. Vol. 1, pp. 22–23.
19. *YOLO9000: Better, Faster, Stronger*. Available at: <https://arxiv.org/pdf/1612.08242.pdf> (accessed 15 April 2019)
20. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. Available at: <https://arxiv.org/abs/1602.07261> (accessed 15 April 2019)
21. Beresnev A.P., Zoev I.V., Markov N.G. Issledovanie svertochnykh neyronnykh setey klassa YOLO dlya mobilnykh setey detektirovaniya obektov na izobrazheniyakh [Research on convolutional neural networks of YOLO class for mobile object detection system]. *Trudy 28-oy mezhdunarodnoy konferentsii* [Graphi-Con 2018. Proc. of 28th International conference on computer graphics and vision]. Tomsk, 2018. pp. 196–199.
22. *Pascal VOC database*. Available at: <https://pjreddie.com/projects/pascal-voc-dataset-mirror/> (accessed 15 April 2019).
23. Morelos-Zaragoza R.H. *The art of error correcting coding. Second edition*. Baffins Lane, Chichester, West Sussex, England, John Willey & Sons, Ltd, 2002. 269 p.
24. Mytsko E.A., Malchukov A.N. Research of software-based implementations of CRC computation algorithms joint with PKZIP, WINRAR, ETHERNET. *Bulletin of the Tomsk Polytechnic University*, 2013, vol. 322, no. 5, pp. 170–175.
25. Ryzhova S.E., Mytsko E.A. Razrabotka adaptivnoy struktury bystrodeystvuyushchego dekodera BCH-koda i ego apparatnaya realizatsiya na primere kodovoy kombinatsii dlinoy v 15 bit na PLIS [Development of adaptive structure of high-speed decoder of BCH-code and its hardware-based implementation on code length example 15 bits using FPGA]. *Nauchnaya sessiya TUSUR-2017: materialy Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii studentov, aspirantov i molodykh uchennykh, posvyashchennoy 55-letiyu TUSURa* [Proc. of International scientific-technical conference of students, graduate students and young scientists for 55-years old of TUSUR: TUSUR Scientific Session 2017]. Tomsk, 2017. Vol. 5, pp. 148–151.
26. *The MNIST database of handwritten digits*. Available at: <http://yann.lecun.com/exdb/mnist> (accessed 26 July 2018).
27. Jia Y., Shelhamer E., J Donahue., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T. Caffe: Convolutional Architecture for Fast Feature Embedding. *Proc. of the 22nd ACM Multimedia*. Orlando, Florida, USA, November 3–7, 2014. pp. 675–678.
28. *Quartus II Handbook*. Vol. 3: Verification. Available at: https://www.altera.com/content/dam/altera-ww/global/en_US/pdfs/literature/hb/qts/qts_qii5v3.pdf (accessed 15 April 2019).
29. *QGIS is a free, open source, cross platform GIS*. Available at: <https://github.com/qgis/QGIS> (accessed 31 January 2019).

Received: 12 June 2019.

Information about the authors

Ivan V. Zoev, post-graduate student, National Research Tomsk Polytechnic University.

Nikolay G. Markov, Dr. Sc., professor, National Research Tomsk Polytechnic University.

Svetlana E. Ryzhova, post-graduate student, National Research Tomsk Polytechnic University.