Исследование свёрточных нейронных сетей класса YOLO для мобильных систем детектирования объектов на изображениях

А.П. Береснев[1], И.В. Зоев[1], Н.Г. Марков[1]
apb3@tpu.ru|ivz3@tpu.ru|markovng@tpu.ru
[1]Томский политехнический университет, г. Томск, Российская Федерация

Предлагаются свёрточные нейронные сети класса YOLO новой архитектуры для решения задачи детектирования объектов на изображениях с помощью аппаратно-реализованной системы компьютерного зрения. Приводятся результаты анализа эффективности таких сетей в сравнении со свёрточными сетями класса YOLO с другими архитектурами.

Ключевые слова: *свёрточные нейронные сети, класс нейронных сетей YOLO, детектирование объектов на изображениях, мобильные системы компьютерного зрения.*

Research on convolutional neural networks of YOLO class for mobile object detection system

A.P. Beresnev[1], I.V. Zoev[1], N.G. Markov[1]
apb3@tpu.ru|ivz3@tpu.ru |markovng@tpu.ru
[1]Tomsk Polytechnic University, Tomsk, Russian Federation

This article presents new convolutional network architectures of the YOLO class to solve the problem of detecting objects in images using a hardware-based computer vision system. The efficiency analysis results of proposed networks in comparison with convolutional networks of YOLO class with other architectures are presented.

Keywords: *convolutional neural network, neural networks of YOLO class, object detection, mobile computer vision systems.*

1. Введение

В последние годы при решении задачи детектирования объектов на изображениях наметилась тенденция к созданию мобильных систем компьютерного зрения (СКЗ) на основе аппаратно-реализованных свёрточных нейронных сетей (СНС) [1]. При этом весьма перспективными являются СНС класса YOLO [7]. Такие СНС обеспечивают довольно высокую точность и скорость детектирования объектов на изображениях, но сложны в аппаратной реализации.

В статье описываются СНС новой архитектуры из класса YOLO, более простые в аппаратной реализации, чем другие сети из этого класса. Приводятся результаты исследования эффективности таких сетей. Дается их сравнение с результатами исследований эффективности СНС других архитектур этого же класса.

2. Задача аппаратной реализации СНС для детектирования объектов на изображениях

В настоящее время во многих областях человеческой деятельности востребованы мобильные (СКЗ), устанавливаемые на беспилотные летательные аппараты (БПЛА). Такие СКЗ должны решать задачи детектирования объектов различной физической природы на изображениях земной поверхности. Под задачей детектирования обычно понимается определение ограничивающих областей и класса для каждого объекта на анализируемых изображениях. При решении задачи детектирования наметилась тенденция к использованию аппаратно-реализованных СНС с целью повышения скорости детектирования объектов. При этом, создание мобильных СКЗ значительно усложняется тем, что при их разработке необходимо искать баланс между точностью детектирования, энергопотреблением и быстродействием бортового вычислительного устройства, реализующего СНС, из-за ограниченности используемых вычислительных ресурсов СКЗ. Для этого требуются дополнительные исследования эффективности разрабатываемых мобильных СКЗ при использовании в них различных классов и архитектур СНС.

Цель данной работы – исследование эффективности различных архитектур СНС для применения в мобильных СКЗ, которые наиболее полно отвечают требованиям по точности детектирования объектов на изображениях, скорости выполнения каждой из архитектур СНС и по сложности её аппаратной реализации. Последние два требования, по сути, определяют требование к быстродействию бортового вычислительного устройства в составе СКЗ.

Анализ показывает, что среди наиболее подходящих для решения задачи детектирования СНС предпочтение сегодня следует отдать сетям класса YOLO [7]. Эти сети обеспечивают большие точность и скорость детектирования объектов на изображениях, чем сети других классов. Проанализируем наиболее известные архитектуры сетей класса YOLO.

Наиболее исследованной и перспективной считается архитектура YOLOv2, включающая 22 свёрточных слоя. В качестве функции активации сетей этого класса используется функция leaky ReLU, определяемая по формуле $max(0{,}1x, x)$. Сеть этой архитектуры имеет ряд отличий от исходной. YOLOv2 – полностью свёрточная сеть (англ. fully convolutional network), из архитектуры этой сети убрано два полносвязных слоя на выходе, что позволяет не использовать метод dropout на этапе обучения для предотвращения переобучения. Чтобы детектировать объекты разного масштаба карты признаков перед последним слоем подвыборки (англ. pooling layer) объединяются с выходными картами признаков выходного слоя. Для определения ограничивающих областей объектов используются смещения относительно якорных значений (англ. anchor boxes), которые вычисляются на из обучащей выборки с использованием метода K средних. Точность

The publication of the conference proceedings was supported by the RFBR, grant No. 18-07-20045\18

детектирования по методике mAP составляет 76,8% на выборке Pascal VOC 2007, а скорость выполнения этой сети при такой точности достигает 67 кадров в секунду на видеокарте NVIDIA Titan X. В силу ограниченности вычислительных ресурсов в мобильных СКЗ перспективной для аппаратной реализации является более простая архитектура tiny-YOLO, имеющая только 9 свёрточных слоёв. В этой архитектуре не используется добавление карт признаков из предыдущих слоев в последующие. Однако точность детектирования и скорость выполнения сети близки к соответствующим значениям в случае использования архитектуры YOLOv2.

Отметим, что во всех архитектурах сетей класса YOLO используется операция батч-нормализации (англ. batch-normalization) после каждого свёрточного слоя. Это усложняет процесс вычислений в сети. Естественно, что при аппаратной реализации СНС для многократного выполнения такой операции должны использоваться дополнительные вычислительные ресурсы и возрастет энергопотребление СКЗ. В этой связи необходимо разрабатывать СНС новой архитектуры без операции батч-нормализации.

3. СНС класса YOLO новой архитектуры

Глубокие нейронные сети были предметом повышенного интереса до недавнего времени. Однако сегодня мобильные устройства позволяют проводить эксперименты по использованию нейронных сетей для решения различного рода задач на самом устройстве. Различные методы позволяют сократить затраты на вычисления и используемые ресурсы, при этом сохраняя точность работы сети в сравнении с более глубокими вариантами сетей. Например, на выборке ImageNet хорошо себя зарекомендовал подход с использованием Inception-модулей, которые применялись в СНС GoogLeNet [4].

В статье [5] приведены результаты исследования, которое показало, что увеличение количество слоёв сети не ведет к увеличению точности детектирования. 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.

Возможна комбинация модулей, когда в Inception-модуль встраивается ResNet-модуль. Такой подход и различные комбинации модулей представлены в статье [5]. В результате сокращается число вычислений в сравнении с более глубокими сетями, но сохраняется точность детектирования. Используем данный подход для развития архитектуры СНС tiny-YOLO, заменив часть слоев на InceptionResNet-модули в различных комбинациях. На рис. 1 представлена детализированная схема Inception-ResNet-модуля.

В табл. 1 представлена исходная архитектура СНС tiny-YOLO. Так как выход InceptionResNet-модуля имеет размер в 256 карт признаков и объединяется со входом, то изменим количество карт признаков четвертого свёрточного слоя с 128 на 256 карт признаков. Также уменьшим количество карт признаков в последующих слоях свёртки сети tiny-YOLO.

Другим важным отличием предлагаемой архитектуры от исходной является отказ от использования операции батч-нормализации.

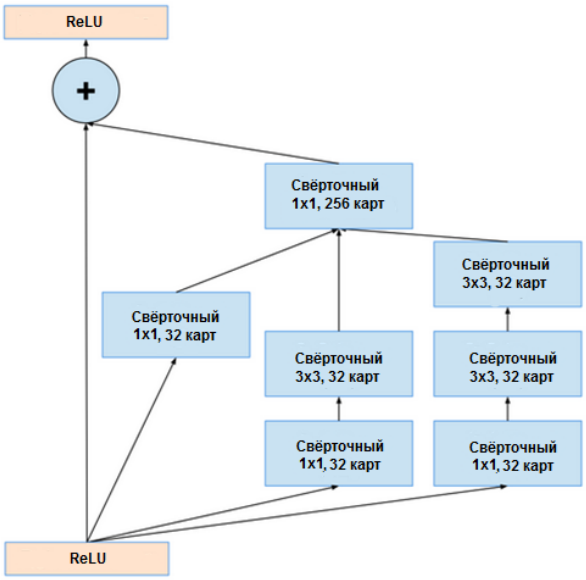

Рисунок 1. Inception-ResNet-модуль

Таблица 1. Архитектура СНС tiny-YOLO

№	Тип слоя	Ядра	Размеры/шаг	Размеры входа
1	Свёрточный	16	3x3/1	416x416x3
2	Подвыборки	-	2x2/2	416x416x16
3	Свёрточный	32	3x3/1	208x208x16
4	Подвыборки	-	2x2/2	208x208x32
5	Свёрточный	64	3x3/1	104x104x32
6	Подвыборки	-	2x2/2	104x104x64
7	Свёрточный	128	3x3/1	52x52x64
8	Подвыборки	-	2x2/2	52x52x128
9	Свёрточный	256	3x3/1	26x26x128
10	Подвыборки	-	2x2/2	26x26x256
11	Свёрточный	512	3x3/1	13x13x256
12	Подвыборки	-	2x2/1	13x13x512
13	Свёрточный	1024	3x3/1	13x13x512
14	Свёрточный	1024	3x3/1	13x13x1024
15	Свёрточный	125	1x1/1	13x13x1024
16	Детектор	-	-	

В итоге в табл. 2 представлена новая архитектура tiny-YOLO-InceptionResnet.

Таблица 2. Архитектура СНС tiny-YOLO-InceptionResNet

№	Тип слоя	Ядра	Размеры/шаг	Размеры входа
1	Свёрточный	16	3x3/1	416x416x3
2	Подвыборки	-	2x2/2	416x416x16
3	Свёрточный	32	3x3/1	208x208x16
4	Подвыборки	-	2x2/2	208x208x32
5	Свёрточный	64	3x3/1	104x104x32

Работа опубликована при финансовой поддержке РФФИ, грант 18-07-20045\18

6	Подвыборки	-	2x2/2	104x104x64
7	Свёрточный	256	3x3/1	52x52x64
8	Подвыборки	-	2x2/2	52x52x256
9	Inception ResNet	-	-	26x26x256
10	Свёрточный	128	3x3/1	26x26x256
11	Подвыборки	-	2x2/2	26x26x128
12	Свёрточный	256	3x3/1	13x13x128
13	Подвыборки	-	2x2/1	13x13x256
14	Свёрточный	128	3x3/1	13x13x256
15	Свёрточный	125	1x1/1	13x13x128
16	Детектор	-	-	

Предложим еще несколько новых архитектур СНС на базе сети tiny-YOLO. Так используем последовательно по два и три InceptionResNet-модуля, которые следуют друг за другом и получим архитектуры tiny-YOLO-InceptionResNet2 и tiny-YOLO-InceptionResNet3. Также параллельно используем два и три таких же модуля, имеющих один вход и конкатенацию их выходных карт признаков, что дает новые архитектуры tiny-YOLO-InceptionResNet-x2 и tiny-YOLO-InceptionResNet-x3.

4. Результаты исследований эффективности сетей класса YOLO

Исследования эффективности СНС YOLOv2, tiny-YOLO и предложенных новых архитектур сетей проводились по скорости выполнения каждой из архитектур, по сложности их для аппаратной реализации и по точности детектирования объектов на тестовой выборке PascalVOC2007[6].

Скорость выполнения каждой из архитектур определялась как среднее время выполнения сети при детектировании объектов одного изображения из тестовой выборки, состоящей из 4952 изображений. В табл. 3 представлены значения среднего времени выполнения для всех исследуемых архитектур СНС. Видим, что наилучшей скоростью обладает архитектура tiny-YOLO-Inception-ResNet , а наихудшей – сеть YOLOv2.

Таблица 3. Среднее время детектирования объектов на одном изображении

Архитектура СНС	Среднее время выполнения СНС, мс
YOLOv2	35,016
tiny-YOLO	12,124
tiny-YOLO-Inception-ResNet	6,943
tiny-YOLO-Inception-ResNet2	7,546
tiny-YOLO-Inception-ResNet3	8,337
tiny-YOLO-Inception-ResNet_x2	7,948
tiny-YOLO-Inception-ResNet_x3	9,047

Сложность при аппаратной реализации СНС определяется числом операций, реализуемых сетью в единицу времени, и объемом требуемой памяти, в первую очередь, для весовых коэффициентов сети. Основные операции, используемые в сетях класса YOLO: умножение с накоплением, сравнение, операции активации, сложение, деление. Причем последние две необходимы при реализации батч-нормализации, которая отсутствует в предложенных новых архитектурах СНС. В табл. 4 в среднем столбце представлено количество всех операций, реализуемых каждой из архитектур. Чем меньше таких операций, тем проще аппаратная реализация архитектуры СНС. В правом столбце приведены значения объема требуемой памяти для весовых коэффициентов СНС.

Таблица 4. Требуемые ресурсы для аппаратной реализации исследуемых архитектур СНС

Архитектура СНС	Число операций СНС, GFlops	Объем весовых коэффициентов СНС, МБ
YOLOv2	29,36	202,7
tiny-YOLO	6,973	63,5
tiny-YOLO-Inception-ResNet	2,451	4,6
tiny-YOLO-Inception-ResNet2	3,136	6,6
tiny-YOLO-Inception-ResNet3	3,622	8,1
tiny-YOLO-Inception-ResNet_x2	3,129	6,6
tiny-YOLO-Inception-ResNet_x3	3,719	8,4

Из табл. 4 следует, что минимальные ресурсы требуются для аппаратной реализации архитектуры tiny-YOLO-Inception-ResNet. Для архитектур YOLOv2 и tiny-YOLO требуется максимальное количество ресурсов, в том числе из-за использования операции батч-нормализации.

Исследование эффективности СНС в части точности детектирования объектов осуществлялось с использованием метрики, применяемой в PascalVOC2007. В PascalVOC2007 точность оценивается интегральным параметром mean Average Precision (mAP). Данный параметр является средним значением от Average Precision (AP) для каждого класса из обучающей выборки. AP рассчитывается по формуле:

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,\ldots,1\}} P(r),$$

$$P = \frac{TP}{TP + FP},$$

$$r = \frac{TP}{TP + TN},$$

где P – precision r – recall, TP – true positive (правильно распознанный объект), FP – false positive (неправильно распознанный объект), FN – false negative (ложное срабатывание детектора на фоне).

Таблица 5. Точность детектирования СНС

Архитектура СНС	mAP, %
YOLOv2	76,80
tiny-YOLO	57,1
tiny-YOLO-Inception-ResNet	46,22
tiny-YOLO-Inception-ResNet2	53,68
tiny-YOLO-Inception-ResNet3	55,45
tiny-YOLO-Inception-ResNet_x2	49,54
tiny-YOLO-Inception-ResNet_x3	46,52

The publication of the conference proceedings was supported by the RFBR, grant No. 18-07-20045\18

В табл. 5 значения mAP для архитектур YOLOv2 и tiny-YOLO взяты из [7].

Нами для обучения СНС с новыми архитектурами использовался open source фреймворк, названный darknet [2]. Он поддерживает декларативное описание архитектуры сетей в виде конфигурационного файла с расширением cfg. При обучении использовались параметры, предлагаемые авторами архитектуры YOLOv2 и tiny-YOLO, однако размер батчей был ограничен 16 элементами обучающей выборки. Обучение предлагаемых архитектур проводилось на видеокарте NVIDIA GTX 1060 на фреймворке darknet с использованием библиотек CUDA 9.1 и cuDNN 7.

Из табл. 5 видно, что наилучшие результаты по точности детектирования даёт СНС YOLOv2, однако использование этой сети, как следует из табл. 3 и табл. 4, требует значительных аппаратных ресурсов и уступает по скорости другим архитектурам. К сожалению, архитектура tiny-YOLO-Inception-ResNet имеет невысокую точность детектирования. Разработчикам мобильных 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.

5. Заключение

В последнее время при решении задачи детектирования объектов на изображениях наметилась тенденция к созданию мобильных СКЗ. В составе таких СКЗ используются аппаратно-реализованные СНС. В данной работе ставилась задача разработки и исследования эффективности различных архитектур СНС класса YOLO для мобильных СКЗ.

Предложены новые архитектуры СНС на основе архитектуры tiny-YOLO без использования батч-нормализации и с включением в архитектуру tiny-YOLO Inception-ResNet-модулей в различных конфигурациях.

Проведенные исследования эффективности известных и предложенных архитектур СНС класса YOLO позволили считать, что наиболее перспективными для аппаратной реализации СНС в составе СКЗ являются предложенные архитектуры tiny-YOLO-Inception-ResNet2 и tiny-YOLO-Inception-ResNet3.

Именно с использованием таких архитектур СНС разработчикам мобильных СКЗ можно найти баланс между точностью и скоростью детектирования объектов на изображениях.

6. Благодарности

Исследования были поддержаны грантом РФФИ № 18-47-700010 р_а.

7. Литература

[1] Зоев, И.В. 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. – DOI: 10.18287/2412-6179-2017-41-6-938-949.

[2] Darknet framework. [Электронный ресурс] – URL: http://github.com/pjreddie/darknet (дата обращения 14.05.2018)

[3] Deep Residual Learning for Image Recognition [Электронный ресурс] – URL: https://arxiv.org/abs/1512.03385 (дата обращения 14.05.2018)

[4] Going Deeper with Convolutions. [Электронный ресурс] – URL: https://arxiv.org/abs/1409.4842 (дата обращения 14.05.2018)

[5] Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. [Электронный ресурс] – URL: https://arxiv.org/abs/1602.07261 (дата обращения 14.05.2018)

[6] Pascal VOC database. [Электронный ресурс] – URL: https://host.robots.ox.ac.uk/pascal/VOC (дата обращения 14.05.2018)

[7] YOLO9000: Better, Faster, Stronger. [Электронный ресурс] – URL: https://arxiv.org/pdf/1612.08242.pdf (дата обращения 14.05.2018)

Об авторах

Береснев Алексей Павлович, магистрант отделения информационных технологий инженерной школы информационных технологий и робототехники Томского политехнического университета, e-mail: apb3@tpu.ru.

Зоев Иван Владимирович, аспирант отделения информационных технологий инженерной школы информационных технологий и робототехники Томского политехнического университета., e-mail: ivz3@tpu.ru.

Марков Николай Григорьевич, д.т.н., профессор отделения информационных технологий инженерной школы информационных технологий и робототехники Томского политехнического университета, e-mail: markovng@tpu.ru.

Работа опубликована при финансовой поддержке РФФИ, грант 18-07-20045\18