УДК 004.032.2
DOI: 10.15827/0236-235X.132.635-640
Дата подачи статьи: 13.05.20
2020. Т. 33. № 4. С. 635–640

Экспериментальный анализ точности и производительности разновидностей архитектур YOLO для задач компьютерного зрения

П.А. Боков [1], *студент, deathhodd@gmail.com*
П.Д. Кравченя [1], *старший преподаватель, kpd_@mail.ru*

[1] *Волгоградский государственный технический университет, г. Волгоград, 400005, Россия*

В повседневную жизнь все активнее внедряются беспилотные автомобили. Для достижения полноценной автономности при поездке они используют системы компьютерного зрения, отвечающие за анализ состояния светофоров, знаков и других объектов, которые могут появляться на дороге. На сегодняшний день стандартом в этой области является архитектура YOLOv1, однако она уже устарела. В связи с этим идет разработка системы компьютерного зрения для беспилотного автомобиля на базе современных технологий.

Существует проблема выбора архитектуры компьютерного зрения, которая будет отвечать за анализ дорожного трафика. В первую очередь, она должна быть быстрой и точной, так как дорожный трафик меняется очень интенсивно, а точность определения напрямую влияет на степень задействованности пассажира-водителя в процессе, чтобы избежать аварийных ситуаций. Помимо этого, архитектура должна занимать как можно меньше вычислительных мощностей и не потреблять большое количество энергоресурсов. Для исследования данных проблем было принято решение о проведении эксперимента, который позволил бы выявить слабые и сильные стороны различных YOLO-архитектур. К тому же данные, предоставляемые исследователями, сильно разнятся из-за использования разного оборудования, на котором осуществляются обучение и тестирование сетей, что не дает возможности использовать их для объективной оценки.

В статье анализируются различные разновидности архитектуры YOLOv3 и ее версии для маломощных вычислительных систем YOLOv3-tiny, описываются их преимущества и недостатки при использовании в системах компьютерного зрения. Эксперименты выполняются на едином оборудовании для всех анализируемых вариантов архитектур. Проводится экспериментальное исследование точности и производительности различных вариантов архитектур YOLO. Для обучения и тестирования используется датасет VOC2012. В результате исследования определяются сильные и слабые стороны рассматриваемых архитектур и анализируются варианты дальнейшего развития технологии с учетом роста мощностей вычислительных систем и появления новых технологических решений.

Ключевые слова: *компьютерное зрение, разработка программного обеспечения, беспилотные автомобили, нейронная сеть, YOLOv3.*

В связи с быстрым развитием беспилотной автомобильной промышленности многие автомобильные и IT-компании выделяют большие средства на поиск новых эффективных решений. Одной из ключевых технологий, используемых в *беспилотных автомобилях* (БПА), является технология распознавания дорожного трафика и позиционирования транспортного средства в пространстве, состоящая из алгоритма SLAM и технологий обнаружения и распознавания объектов. SLAM используется в мобильных автономных средствах для построения карты в неизвестном пространстве или для обновления карты в заранее известном пространстве с одновременным контролем текущего местоположения и пройденного пути [1]. Однако использование только одного SLAM неэффективно, так как дорожное движение регулируется указателями, дорожными знаками и светофорами. Данную проблему решают программно-аппаратные средства детектирования объектов.

Стандартом в области распознавания систем контроля за дорожным трафиком стала архитектура YOLOv1 [2], которая на момент своего появления была самым перспективным решением и характеризуется довольно высокой скоростью и точностью детектирования. Помимо этого, данная архитектура хорошо распознает небольшие объекты, что очень важно.

Успешность архитектуры дала толчок к ее развитию, что привело к появлению новых вер-

сий, учитывающих проблемы предшественников и реализующих новые решения. Использование первой версии архитектуры обусловлено тем, что переводить систему на новые архитектурные решения невыгодно.

Анализ рынка решений для детектирования объектов показал, что открытых систем в наличии нет, все они – частная собственность компаний-разработчиков БПА и крупнейших производителей автомобилей. Исключением являются патенты компании Tesla, которая открыла их для использования [3]. В связи с этим разрабатываемая система будет базироваться на самой новой версии алгоритма YOLOv3, в том числе быстро и точно распознавать системы контроля за дорожным трафиком.

Усовершенствованная версия YOLOv3 включает в себя вариации 320, 416 и 608, что соответствует размеру изображения, к которому оно будет приведено перед отправкой на вход нейросети, по высоте и ширине. YOLOv3 состоит из 106 сверточных слоев, графически представленных на рисунке 1. Изображение на входе пропускается через модифицированную архитектуру GoogLeNet [4], и после каждого сверточного слоя происходит процесс батч-нормализации [5]. На выходе были убраны два полносвязных слоя, что позволило отказаться от использования метода выбывания, предназначенного для предотвращения переобучения в процессе обучения сети, но имеются три слоя для определения изображения разных размеров и перед последним слоем подвыборки карты признаков объединяются с картами признаков выходного слоя [6].

Основным отличием от других *сверточных нейронных сетей* (СНС) является применение сети ко всему изображению сразу, а не несколько раз к разным регионам. Архитектура YOLO формирует сетку по изображению и предсказывает положение bounding boxes (bbox) [7], а также вероятность ячейки содержать нужное изображение. Достигается это с помощью определения границ методом среднего K, который, в свою очередь, использует базовые значения bbox для каждого класса, и умножением вероятности ячейки на вероятность каждого класса, что в итоге и показывает вероятность содержания в области объекта. Помимо этого, при любом значении границ bbox центральной точкой будет центр классифицируемого объекта.

YOLOv3-tiny представляет собой упрощенную архитектуру, предназначенную для ра-

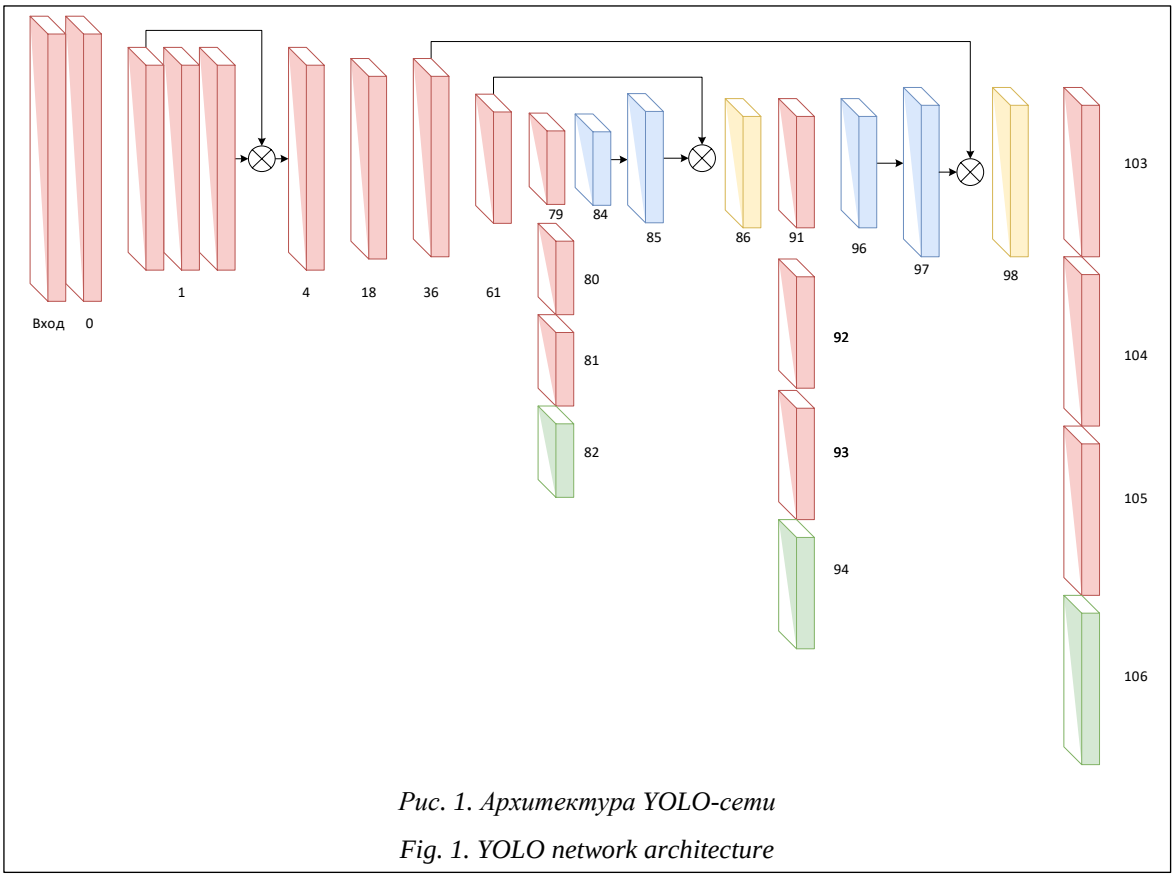

Рис. 1. Архитектура YOLO-сети

Fig. 1. YOLO network architecture

боты на маломощных устройствах и небольших датасетах. В ней количество сверточных слоев уменьшено со 106 до 16 и отсутствует объединение пар признаков. При этом существенно повышается производительность, но сильно теряется точность.

Точность определения по методике mAP составляет 51,5 для датасета COCO (Common Objects in Context) при мощности в 38,97 ГФлопс для архитектуры YOLOv3-320 и 33,1 при мощности 5,56 ГФлопс для архитектуры tiny, что является лучшим результатом в соотношении точность–производительность, делая вариант архитектуры 320 лучшим из доступных вариаций архитектуры YOLOv3 [8]. Частота кадров при таком соотношении для YOLOv3 составляет 35, а для tiny – 220. Данные значения показывают сильный разрыв в значениях у архитектур v3 в отличие от второй версии, где различия в точности и скорости были значительно меньше.

YOLOv3 является ресурсоемкой архитектурой. Один из ключевых моментов, который на это сильно влияет, – наличие батч-оптимизации. Так как данная операция выполняется перед каждым сверточным слоем и сама по себе является довольно ресурсоемкой и энергозатратной, общее потребление ресурсов системой компьютерного зрения сильно возрастает. Учитывая ограниченность доступных мощностей, отказ от использования оптимизации существенным образов увеличит скорость определения.

Наличие в YOLOv3 модифицированной GoogLeNet-архитектуры дает возможность использовать хорошо показавшие себя Inception-модули, которые позволяют увеличить производительность без потери скорости и точности. В работе [9] отмечается, что общее увеличение количества сверточных слоев не способствует увеличению точности. Для решения этой задачи была предложена архитектура ResNet, характеризующаяся передачей данных со входа слоя на следующий и через два слоя. Данная операция позволяет уменьшить количество слоев, при этом не потеряв точности, а в некоторых ситуациях даже увеличив ее. В этой же работе рассматриваются различные комбинации модулей (например, совмещение Inception-модулей и архитектуры ResNet), которые позволяют сократить количество вычислений и при этом сохранить точность детектирования. На основе результатов данных исследований было принято решение модифицировать архитектуру YOLOv3-tiny с целью получения информации о перспективности ее использования в задачах компьютерного зрения. Для этого количество карт признаков для пятисверточного модуля было увеличено в два раза – со 128 до 256, а во всех остальных слоях свертки уменьшено. Это обусловлено использованием в ResNetInception на выходе 256 карт. Интеграция данных модулей была отражена в названии полученной модифицированной архитектуры – YOLOv3-tiny-ResNetInception.

Для увеличения производительности YOLOv3 хорошим решением является использование архитектуры XNOR-net, предложенной в статье [10]. Ее особенность заключается в аппроксимации конволюционных вычислений с использованием в основном двоичных операций. Графическое представление XNOR-net дано на рисунке 2.

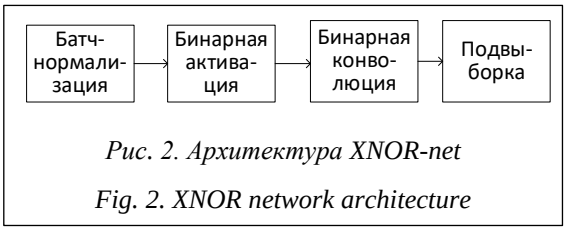

Рис. 2. Архитектура XNOR-net

Fig. 2. XNOR network architecture

Для проверки эффективности оригинальных и модифицированных архитектур YOLO была использована реализация фреймворка darknet с поддержкой архитектуры XNOR-net. Во фреймворке уже реализованы некоторые модели с использованием XNOR, однако данные по их скорости и точности отсутствуют. Поэтому было принято решение провести тестирование архитектуры YOLOv3-XNOR.

Для обучения и тестирования выбран датасет VOC2012. В тест вошли и оригинальные архитектуры, так как VOC содержит большее количество дорожных знаков не в самом лучшем разрешении, что дает более близкий к реальным условиям результат. Тестирование на стенде оригинальных архитектур обусловлено необходимостью получения объективных данных при выполнении моделирования на вычислительном устройстве, которое используется и для тестирования модифицированных версий.

Целью теста являлось получение результатов по точности, времени детектирования объектов и сложности аппаратной реализации архитектур (см. таблицу). Использован следующий стенд: ЦП – QuadCore AMD Ryzen 5 1500X, 3 600 MHz (36×100); ОЗУ – 12 Гб; ГПУ – GeForce GTX 1060 6 GB; ОС – Microsoft Windows 10 Home.

Результаты тестирования

Test results

Архитектура	Среднее время выполнения, мс	Число операций, ГФлопс	mAP, %
YOLOv3-320	25,148±0,5	36,04±0,7	78,5±1
YOLOv3-tiny	11,033±0,3	8,01±0,4	52,8±2
YOLOv3-XNOR	21,003±0,5	29,46±0,3	78,7±1
YOLOv3-tiny-ResNetInception	7,459±0,2	6,72±0,1	50,9±3

Среднее время выполнения определялось временем, которое затрачивала сеть на детектирование объектов одного изображения из тестового датасета, состоящего из 2 000 изображений. Число операций отображает количество операций умножения, сравнения, активации, сложения и деления за единицу времени, которые выполняла сеть. Для оценки точности использовался метод с параметром mean Average Precision (mAP), реализованный в VOC2012. Данный параметр является средним значением от Average Precision (AP) для каждого класса из обучающей выборки. AP рассчитывается по формуле

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

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

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).

Результаты тестирования наглядно демонстрируют слабые и сильные стороны архитектур. Так, YOLOv3 и YOLOv3-XNOR показывают хорошие результаты точности в детектировании объектов, но при этом обе остаются такими же требовательными к вычислительным мощностям. Тем не менее, модифицированная архитектура получила прирост в скорости. В свою очередь, YOLOv3-tiny и YOLOv3-tiny-ResNetInception показывают высокую скорость работы, но имеют невысокую точность. Модифицированная архитектура получила прирост в скорости, но при этом потеряла в точности. Такая потеря обусловлена используемым датасетом, который содержит больше мелких объектов, и проблемой в детектировании таких объектов tiny-архитектурой.

Решить проблему со скоростью вычислений могут современные нейронные процессоры. Они являются специализированным классом микропроцессоров и сопроцессоров (часто специализированной интегральной схемой), используемых для аппаратного ускорения работы алгоритмов искусственных нейронных сетей, компьютерного зрения, распознавания по голосу, машинного обучения и других методов искусственного интеллекта [11]. Проблема с точностью у архитектуры tiny остается пока нерешенной. Использовать ее в беспилотном автомобиле нельзя, так как она плохо распознает небольшие объекты и имеет низкую точность, что может привести к аварийной ситуации или к неправильному построению маршрута. Tiny-архитектуры преимущественно используются на устройствах с низкой вычислительной мощностью, в задачах, не сопряженных с риском для человеческой жизни.

Заключение

Результаты тестирования оригинальных и модифицированных архитектур YOLO показали, что вариация YOLOv3 с архитектурой XNOR-net является самой перспективной и наиболее подходящей к задачам компьютерного зрения в БПА. Архитектура YOLOv3-XNOR продемонстрировала наибольшую точность, к тому же имеет более высокую скорость работы по сравнению с оригинальной YOLOv3. YOLOv3-tiny и модифицированная архитектура с использованием Inceprion и ResNet-модулей показали средние значения точности и не могут быть использованы в БПА из-за риска для жизни пассажиров и водителя.

Дальнейшее развитие БПА и их популяризация дадут необходимый толчок для удешевления и развития нейронных процессоров, в том числе для решения проблем, связанных с энергообеспечением систем компьютерного зрения.

Для разработки системы компьютерного зрения для БПА было принято решение использовать XNOR-net как наиболее точную и в долгосрочной перспективе самую оптимальную архитектуру.

Литература

1. RoboCraft. SLAM. URL: http://robocraft.ru/blog/technology/724.html (дата обращения: 29.04.2020).
2. Review: YOLOv1 – You Only Look Once (Object Detection). URL: https://towardsdatascience.com/

yolov1-you-only-look-once-object-detection-e1f3ffec8a89 (дата обращения: 29.04.2020).

3. Tesla's Free-to-Use Patents are All about Sustainability and Strength. URL: https://thedriven.io/2019/02/04/tesla-patents-free-to-use-sustainable-strength/ (дата обращения: 8.05.2020).

4. GoogleNet. URL: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/googlenet.html?q= (дата обращения: 01.05.2020).

5. Batch-normalization. URL: http://neerc.ifmo.ru/wiki/index.php?title=Batch-normalization (дата обращения: 02.05.2020).

6. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. 2018. URL: https://arxiv.org/abs/1804.02767 (дата обращения: 10.05.2020).

7. Convex Hull, (Minimum) Bounding Box, and Minimum Enclosing Circle. URL: http://dwoll.de/rexrepos/posts/diagBounding.html (дата обращения: 02.05.2020).

8. YOLO: Real-Time Object Detection. URL: https://pjreddie.com/darknet/yolo/ (дата обращения: 3.05.2020).

9. Szegedy C., Loffe S., Vanhoucke V., Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. 2016. URL: https://arxiv.org/abs/1602.07261v2 (дата обращения: 10.05.2020).

10. Chowdhury A.P., Kulkarni P., Bojnordi M.N. MB-CNN: Memristive binary convolutional neural networks for embedded mobile devices. Journal of Low Power Electronics and Applications, 2018, no. 8, p. 38. DOI: 10.3390/JLPEA8040038.

11. Neural Processing Unit. URL: https://www.pcmag.com/encyclopedia/term/neural-processing-unit (дата обращения: 3.05.2020).

Software & Systems
DOI: 10.15827/0236-235X.132.635-640

Received 13.05.20
2020, vol. 33, no. 4, pp. 635–640

Experimental analysis of the accuracy and performance of varieties of YOLO architectures for computer vision problems

P.A. Bokov [1], *Student, deathhodd@gmail.com*
P.D. Kravchenya [1], *Senior Lecturer, kpd_@mail.ru*

[1] *Volgograd State Technical University, Volgograd, 400005, Russian Federation*

Abstract. Unmanned vehicles are increasingly being introduced into everyday life. To achieve full autonomy when traveling, unmanned vehicles use computer vision systems, which are responsible for analyzing the status of traffic lights, signs, and other objects that can appear on the road. Today, the standard in this area is YOLOv1 architecture, however, it is already obsolete. In this regard, a computer vision system for an unmanned vehicle based on modern technologies is being developed.

There is the problem of choosing a computer vision architecture that will be responsible for analyzing traffic. First of all, it must be fast and accurate, because road traffic changes very quickly, and the accuracy of determination directly affects the degree of involvement of passenger-drivers in the process in order to avoid emergency situations. In addition, the architecture should occupy as little computing power as possible, and not waste a large number of energy resources. To investigate these issues, it was decided to carry out an experiment that would reveal the advantages and disadvantages of various YOLO architectures. Also, the data provided by different researchers is very different due to using different equipment while training and testing of networks. This makes it impossible for data to be compared objectively.

The paper analyzes various types of YOLOv3 architecture and its versions for low-power computing systems YOLOv3-tiny, describes their advantages and disadvantages for computer vision systems. The experiments are carried out on single hardware for all analyzed architectures. Experimental research on the accuracy and performance of various YOLO architectures is being done. The VOC2012 dataset is used for training and testing. As a result of the research, the strengths and weaknesses of the architectures under consideration are determined and options for the further development of the technology are analyzed, taking into account the growth in the power of computing systems and the emergence of new technological solutions.

Keywords: computer vision, software development, unmanned vehicles, neural network, YOLOv3.

References

1. *RoboCraft. SLAM.* Available at: http://robocraft.ru/blog/technology/724.html (accessed April 29, 2020).

2. *Review: YOLOv1 – You Only Look Once (Object Detection)*. Available at: https://towardsdatascience.com/yolov1-you-only-look-once-object-detection-e1f3ffec8a89 (accessed April 29, 2020).

3. *Tesla's Free-to-Use Patents are All about Sustainability – and Strength*. Available at: https://thedriven.io/2019/02/04/tesla-patents-free-to-use-sustainable-strength/ (accessed May 08, 2020).

4. *GoogleNet*. Available at: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/googlenet.html?q= (accessed May 01, 2020).

5. *Batch-normalization.* Available at: http://neerc.ifmo.ru/wiki/index.php?title=Batch-normalization (accessed May 02, 2020).

6. Redmon J., Farhadi A. *YOLOv3: An Incremental Improvement*. 2018. Available at: https://arxiv.org/abs/1804.02767 (accessed May 10, 2020).

7. *Convex Hull, (Minimum) Bounding Box, and Minimum Enclosing Circle*. Available at: http://dwoll.de/rexrepos/posts/diagBounding.html (accessed May 02, 2020).

8. *YOLO: Real-Time Object Detection*. Available at: https://pjreddie.com/darknet/yolo/ (accessed May 03, 2020).

9. Szegedy C., Loffe S., Vanhoucke V., Alemi A. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016. Available at: https://arxiv.org/abs/1602.07261v2 (accessed May 10, 2020).

10. Chowdhury A.P., Kulkarni P., Bojnordi M.N. MB-CNN: Memristive binary convolutional neural networks for embedded mobile devices. *Journal of Low Power Electronics and Applications*, 2018, no. 8, p. 38. DOI: 10.3390/JLPEA8040038.

11. *Neural Processing Unit*. Available at: https://www.pcmag.com/encyclopedia/term/neural-processing-unit (accessed May 03, 2020).

Для цитирования

Боков П.А., Кравченя П.Д. Экспериментальный анализ точности и производительности разновидностей архитектур YOLO для задач компьютерного зрения // Программные продукты и системы. 2020. Т. 33. № 4. С. 635–640. DOI: 10.15827/0236-235X.132.635-640.

For citation

Bokov P.A., Kravchenya P.D. Experimental analysis of the accuracy and performance of varieties of YOLO architectures for computer vision problems. *Software & Systems*, 2020, vol. 33, no. 4, pp. 635–640 (in Russ.). DOI: 10.15827/0236-235X.132.635-640.