УДК 004.932.2

СОВРЕМЕННЫЕ СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ ОБЪЕКТОВ

Ерохин Д.Ю., *аспирант Рязанского государственного радиотехнического университета (РГРТУ), e-mail: erokhin.d.y@gmail.com;*
Ершов М.Д., *аспирант, ассистент кафедры автоматики и информационных технологий в управлении РГРТУ, e-mail: aitu@rsreu.ru.*

MODERN CONVOLUTIONAL NEURAL NETWORKS FOR OBJECT DETECTION AND RECOGNITION

Erokhin D.Y., Ershov M.D.

This paper contains a comparison of different neural network architectures that are used to solve the problem of object detection and recognition. Modern artificial neural networks are able to detect and localize objects of known classes. This allows them to be used in various technical vision systems. In this work we compare three architectures (YOLO, Faster R-CNN, SSD) by the following criteria: processing speed, mAP, precision and recall.

Key words: intelligent systems, image processing, object detection, pattern recognition, neural networks, machine learning.

Ключевые слова: интеллектуальные системы, обработка изображений, обнаружение объектов, распознавание образов, нейронные сети, машинное обучение.

Проводится сравнение различных нейросетевых архитектур для решения задачи обнаружения и распознавания объектов. Современные искусственные нейронные сети способны обнаруживать и локализовывать объекты заранее известных классов, что позволяет их применять в различных системах технического зрения. Проведено сравнение трех архитектур (YOLO, Faster R-CNN, SSD) по следующим критериям: скорость обработки изображения, mAP, точность и полнота.

Введение

Интеллектуальные системы обработки видеоинформации широко используются в настоящее время в различных областях жизни человека [1, 2]. Развитие подобных систем, с одной стороны, связано с успехами в области вычислительной техники, а с другой – с развитием теории и методов обработки и анализа видеоизображений. Ключевыми задачами, которые лежат в основе большинства приложений систем обработки видеоинформации, являются обнаружение, распознавание и прослеживание объектов [3]. Решение этих задач лежит в основе таких приложений, как системы автоматического обнаружения и сопровождения объектов, робототехника, охранные системы, системы видеоаналитики [4].

В данной работе рассматривается проблема применения искусственных сверточных нейронных сетей в задаче обнаружения и локализации объектов заданных классов.

До появления специализированных нейросетей для обнаружения объектов заданных классов обычно использовался подход, согласно которому изображение проходилось скользящим окном, для каждого положения окна вычислялась карта признаков, например, с помощью гистограммы направленных градиентов или предварительно обученной нейронной сети, которая в последующем поступала на какой-либо классификатор [5]. В качестве классификатора, например, мог использоваться классификатор на основе машин опорных векторов (SVM – support vector machine) [6].

В настоящее время задачу обнаружения и классификации объектов принято решать с помощью искусственных сверточных нейронных сетей. Это обусловлено рядом причин:

– существенный прогресс в области создания графических процессоров;

– большой объем данных для обучения;

– лучшие результаты по сравнению с классическими подходами;

– большое количество специализированных программных пакетов для подготовки данных, обучения и использования нейронных сетей.

Нейросетевые архитектуры для обнаружения и распознавания объектов можно разделить на две большие группы:

1. Архитектуры, обрабатывающие регионы на изображении (R-CNN).

2. Архитектуры, обрабатывающие поступившее изображение целиком (YOLO, SSD).

Архитектура YOLO (You Only Look Once)

В работах [7, 8, 9] представлена архитектура нейросетевого детектора объектов под названием YOLO и ее модификации. Архитектура YOLO изначально разрабатывалась для задач реального времени. В алгоритме YOLO изображение разделяется на ячейки с использованием сетки. Для каждой ячейки сетки оценивается вероятность присутствия объекта вообще, затем строятся несколько наиболее вероятных положений объекта в виде прямоугольников с центром в данной ячейке,

после чего для каждого полученного прямоугольника выполняется оценка вероятностей наличия в нем объектов каждого рассматриваемого класса.

В методе YOLO результаты обнаружения представляются в виде тензора размером 7×7×1024. Оценка вероятности нахождения объекта конкретного класса в текущем обрамляющем прямоугольнике – это произведение оценки вероятности нахождения объекта в ячейке и оценки вероятности для конкретного класса.

В случае YOLOv3 [9] для выделения признаков используется сверточная нейронная сеть, которая состоит из 53 слоев, в качестве фильтров используются свертки размером 3×3 и 1×1 и Residual блоки, которые добавляют к выходу текущего слоя значения с выхода предыдущего слоя.

Также стоит отметить, что в YOLOv3 обнаружение объектов выполняется на трех масштабах, что позволило увеличить качество обнаружения небольших объектов. Сеть масштабирует входное изображение пока не достигнет первого уровня обнаружения, на этом этапе шаг фильтров равен 32-м. На последующих сверточных слоях шаг фильтров равен 2. На каждом масштабе обнаружения ячейка предсказывает три обрамляющих прямоугольника, то есть с учетом масштаба каждой ячейке соответствует 9 обрамляющих прямоугольников.

На следующем шаге выполняется фильтрация прямоугольников по вероятности нахождения в них объектов. Потом так же, как и в архитектуре SSD прямоугольники фильтруются с помощью алгоритма подавления ложных максимумов.

Известно, что большинство алгоритмов распознавания предполагают, что выходные метки являются взаимоисключающими. В архитектурах YOLOv1 [7] и YOLOv2 [8] применяют функцию softmax [10] для преобразования оценок в вероятности классов, суммирование которых по всем классам дает единицу. YOLOv3 использует классификацию с несколькими метками. Например, выходные метки могут быть «Пешеход» и «Ребенок», которые не являются взаимоисключающими и сумма выходов может быть больше 1. В YOLOv3 функция активации softmax заменяется на независимые логистические классификаторы для вычисления вероятности выхода, принадлежащей определенной метке. Вместо использования среднеквадратической ошибки при вычислении потери классификации YOLOv3 использует бинарную кросс-энтропий-ную функцию потерь, вычисляемую для каждого класса. Использование этой техники также позволяет сократить объем требуемых вычислений.

Архитектура Faster R-CNN (Faster Region-based Convolution Neural Network)

Для решения задачи обнаружения объектов Faster R-CNN [11] в настоящее время является одной из часто используемых архитектур на основе глубокого обучения. Предшественниками данной архитектуры являются R-CNN [12] и Fast R-CNN [13].

Работа R-CNN состоит из трех основных этапов:

1. Исходное изображение разбивается на регионы, в которых могут находиться объекты. С этой целью применяется алгоритм Selective Search [14], генерирующий 2000 различных областей, которые с наибольшей вероятностью содержат объекты.

2. Каждый регион подается на вход соответствующей обученной сверточной нейронной сети, которая извлекает вектор признаков для своего региона.

3. Вектора признаков подаются на вход набора SVM, выполняющих функцию классификации. Каждая SVM обучена для определения одного класса объектов. Кроме того, для уточнения параметров охватывающего объект прямоугольника применяется линейная регрессия.

Дополнительным шагом можно считать подавление немаксимумов (алгоритм non-maximum suppression) для исключения избыточного числа прямоугольников, охватывающих один и тот же объект.

Архитектура R-CNN показала высокие показатели точности обнаружения объектов, но были отмечены такие недостатки, как высокие затраты памяти и времени на обучение и обработку изображений. Поэтому были предложены модификации архитектуры, приведшие к созданию Fast R-CNN:

1. Исходное изображение целиком подается на вход одной сверточной нейронной сети, выполняющей извлечение признаков, и на основании полноразмерной карты признаков осуществляется выбор регионов-кандидатов.

2. Набор SVM, выполняющих функцию классификации, был заменен слоем softmax.

Таким образом, сверточная нейронная сеть используется один раз для всего изображения вместо обработки 2000 пересекающихся областей, также достаточно обучить одну сеть со слоем softmax без дополнительного обучения множества SVM.

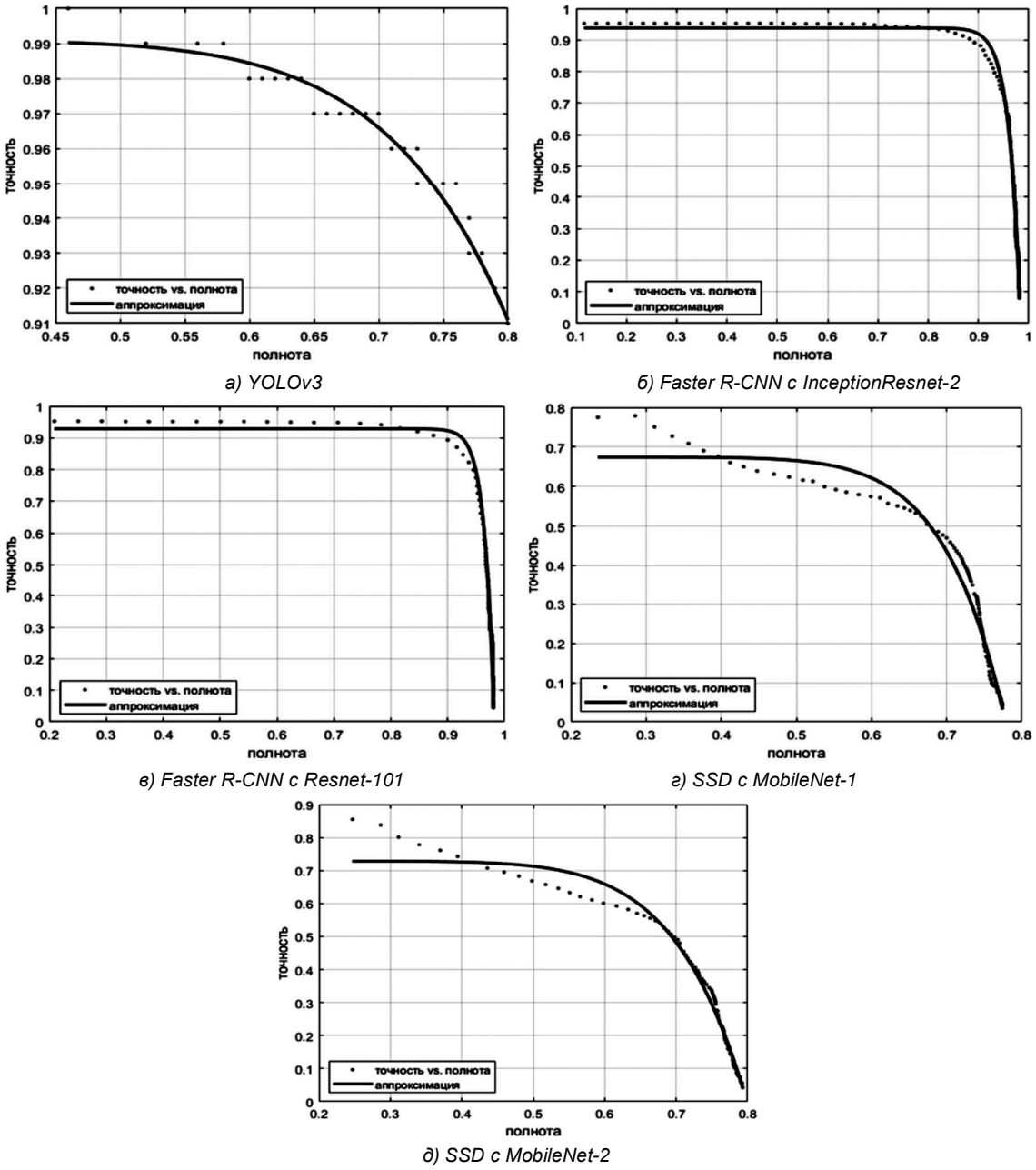
С точки зрения скорости метод Fast R-CNN имеет значительное преимущество перед R-CNN, но еще одним недостатком являлся алгоритм выбора регионов-кандидатов (Selective Search). Модификация данного этапа привела к созданию Faster R-CNN.

Алгоритм Selective Search был заменен на сеть выбора регионов-кандидатов (RPN – region proposal network). На вход данной сети подается область размера n×n, взятая из полноразмерной карты признаков, результат передается на два полносвязных слоя: box-regression и box-classification. Регионы-кандидаты, полученные с помощью RPN, представлены координатами описывающего прямоугольника и вероятностью нахождения объекта в данном регионе, вычисленной с применением функции softmax.

Архитектура Faster R-CNN в настоящее время позволяет добиться высокой точности обнаружения объектов и считается относительно быстрой. При этом сохранена главная идея исходной архитектуры R-CNN: выделение на изображении регионов, в которых возможно находятся объекты, и классификация содержимого этих регионов.

Архитектура SSD (Single Shot MultiBox Detector)

Архитектура SSD [15] обеспечивает значительный прирост скорости обработки по сравнению с Faster R-CNN. Если последняя выполняет выбор регионов-



а) YOLOv3

б) Faster R-CNN с InceptionResnet-2

в) Faster R-CNN с Resnet-101

г) SSD с MobileNet-1

д) SSD с MobileNet-2

Рис. 1. Зависимость точности от полноты

кандидатов и классификацию регионов в два отдельных этапа, то SSD выполняет эти действия одновременно при обработке всего изображения. Работу SSD можно описать следующим образом:

1. Исходное изображение проходит через ряд сверточных слоев, что в результате дает набор карт признаков для разных масштабов (например, 19×19, 10×10, 5×5 и т.д.).

2. В каждой точке каждой карты признаков применяется сверточный фильтр размера 3×3 для получения множества описывающих прямоугольников.

3. Для каждого прямоугольника одновременно оцениваются пространственное смещение и вероятность нахождения объекта.

4. В процессе обучения истинные описывающие объект прямоугольники сопоставляются с предсказанными для исключения ложных обнаружений.

В отличие от R-CNN, где в регионах-кандидатах имеется хотя бы минимальная вероятность нахождения объекта, в SSD шаг фильтрации регионов отсутствует. В результате генерируется гораздо большее количество описывающих прямоугольников на разных масштабах по сравнению с R-CNN, и большая часть не содержит объект. С целью решения данной проблемы в SSD, во-первых, используется подавление немаксимумов для объединения похожих друг на друга прямоугольников в один. Во-вторых, используется техника hard negative mining [16], согласно которой на каждой итерации обучения используется только часть отрицательных примеров, в SSD отношение числа отрицательных примеров к положительным равно 3 к 1.

Выбор регионов-кандидатов и классификация выполняются одновременно: при заданном числе классов C каждый описывающий прямоугольник связан с (4+C)-мерным вектором, который содержит 4 координаты и вероятности для всех классов. На последнем этапе применяется функция softmax для классификации объектов.

Экспериментальные исследования

С целью проведения сравнения было обучено пять нейросетевых детекторов:

1. YOLOv3.

2. Faster R-CNN на базе сети InceptionResnet-2, используемой для извлечения признаков.

3. Faster R-CNN на базе сети Resnet-101, используемой для извлечения признаков.

4. SSD на базе сети MobileNet-1, используемой для извлечения признаков.

5. SSD на базе сети MobileNet-2, используемой для извлечения признаков.

При обучении использовалось около 6700 изображений с размеченными объектами классов «пешеход» и «автомобиль».

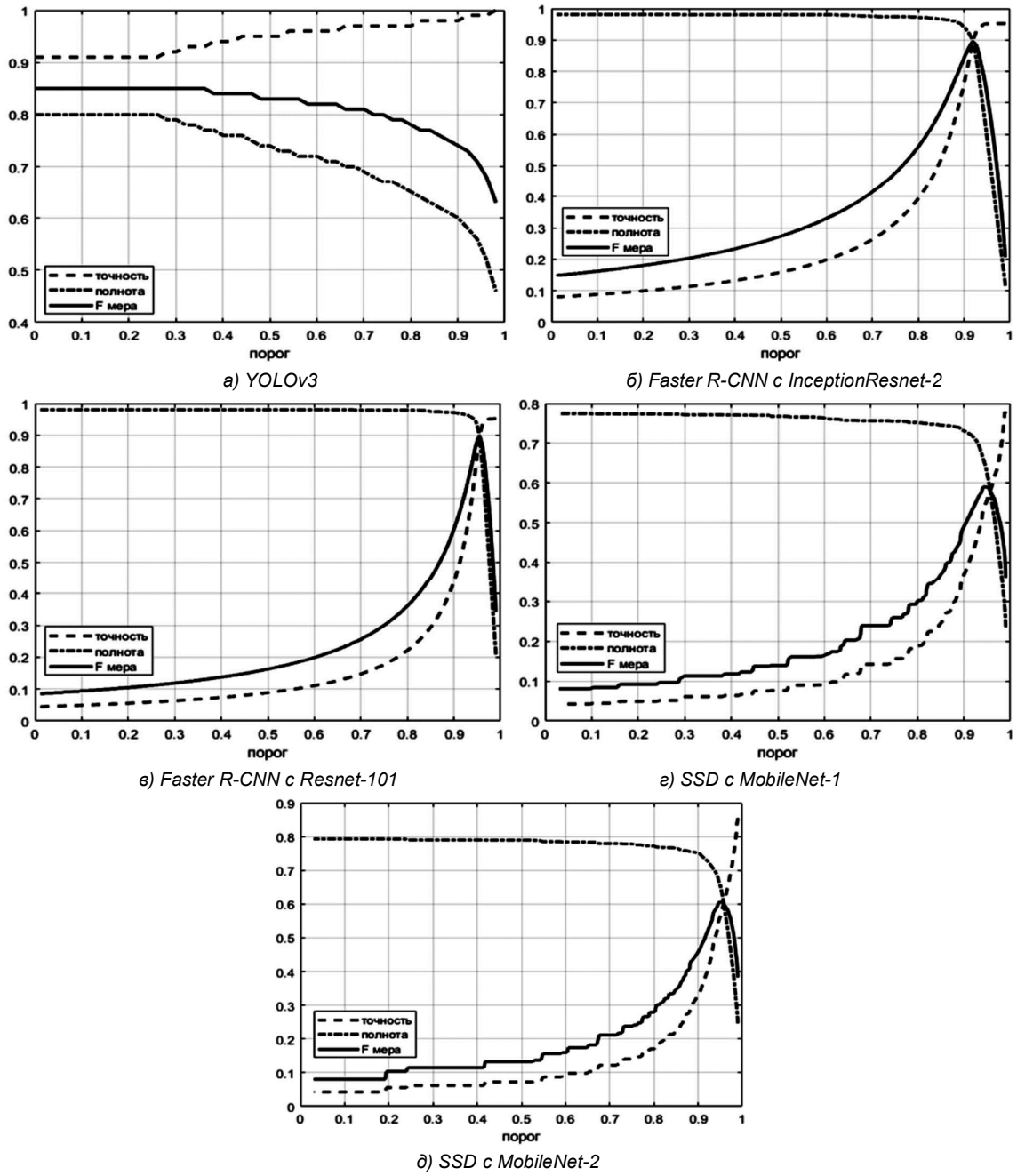
Оценить качество детектора объектов можно путем построения графика зависимости точности (precision) от полноты (recall), а также графиков зависимостей точности, полноты и F-меры от заданного порога. Точность, полнота и F-мера рассчитываются по формулам:

$$precision = \frac{n_{TP}}{n_{TP} + n_{FP}},$$

$$recall = \frac{n_{TP}}{n_{TP} + n_{FN}},$$

$$F = 2\frac{precision \cdot recall}{precision + recall},$$

где n_{TP} (True Positive) – число верно обнаруженных объектов заданного класса; n_{FP} (False Positive) – число ложных срабатываний; n_{FN} (False Negative) – число необнаруженных объектов (пропусков).



а) YOLOv3

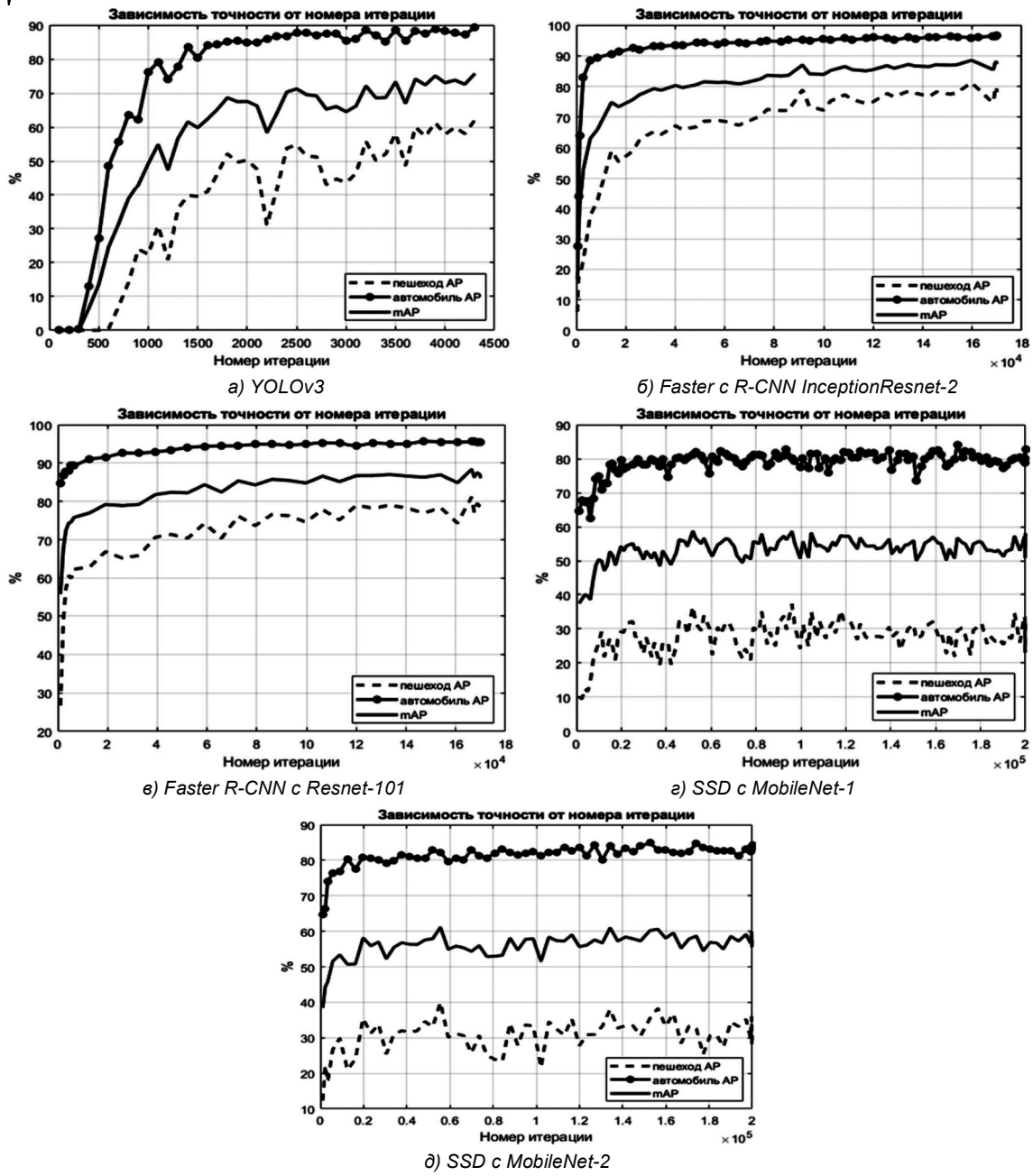
б) Faster R-CNN с InceptionResnet-2

в) Faster R-CNN с Resnet-101

г) SSD с MobileNet-1

д) SSD с MobileNet-2

Рис. 2. Зависимости точности, полноты и F-меры от порога



а) YOLOv3

б) Faster с R-CNN InceptionResnet-2

в) Faster R-CNN с Resnet-101

г) SSD с MobileNet-1

д) SSD с MobileNet-2

Рис. 3. Зависимости AP и mAP от номера итерации обучения

При проведении эксперимента обрабатывалось 750 изображений, содержащих объекты классов «пешеход» и «автомобиль». Для построения графиков изменяется пороговый коэффициент в алгоритме обнаружения объектов в диапазоне от 0 до 1 с шагом в 0,01. Под пороговым коэффициентом понимается минимальное значение оценки вероятности, при котором будет принято решение об обнаружении объекта. На рис. 1 представлены графики зависимости точности от полноты для разных детекторов, на рис. 2 – графики зависимостей точности, полноты и F-меры от заданного порога.

Для оценки качества работы детекторов в зависимости от номера итерации обучения для каждого класса объектов производился расчет метрики AP (Average Precision) – среднего значения максимальной точности при разных значениях полноты. Графики для разных детекторов представлены на рис. 3. Также на графиках отображена метрика mAP (mean Average Precision), представляющая собой среднее значение AP по всем классам объектов.

В качестве интегральных оценок качества работы детекторов использовались площадь под графиком зависимости точности от полноты (AUC – area under curve) и mAP. Также была проведена оценка вычислительной эффективности каждого детектора, при этом использовалась персональная ЭВМ с графическим процессором NVIDIA GeForce GTX 1070, замерялось среднее время обработки кадра с разрешением 720×468. Названные характеристики представлены в табл. 1.

Следует отметить, что время обработки сильно зависит от конкретной конфигурации оборудования, поэтому результаты несут информативный характер. Также время обработки с использованием графических процессоров персональных ЭВМ не всегда отражает время работы на мобильном устройстве. Например, MobileNet-2 на мобильных устройствах работает быст-

рее, чем MobileNet-1, но на персональной ЭВМ небольшое преимущество наоборот получила первая версия данной нейронной сети.

Таблица 1 – Характеристики обученных детекторов

Детектор	Характеристики		
	AUC	*mAP, %*	*Время, мс*
SSD с MobileNet-1	0,534	57,204	56
SSD с MobileNet-2	0,573	61,249	58
YOLOv3	0,882	75,740	76
Faster R-CNN с Resnet-101	0,695	86,411	89
Faster R-CNN с InceptionResnet-2	0,722	88,376	119

На рис. 4 приведены примеры обрабатываемых изображений с обнаруженными объектами интереса (автомобили и пешеходы), использовались наборы данных KITTI [17] и Cityscapes [18].

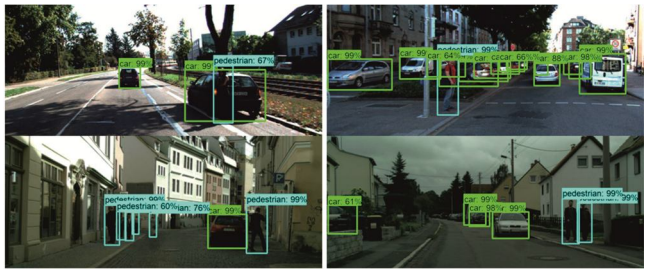


Рис. 4. Примеры изображений с обнаруженными объектами

Заключение

В данной работе рассмотрены три нейросетевые архитектуры: R-CNN (обработка регионов на изображении), YOLO и SSD (обработка поступившего изображения целиком). Экспериментальные исследования качества и вычислительной эффективности проводились с использованием таких нейросетевых детекторов, как YOLOv3, Faster R-CNN с InceptionResnet-2, Faster R-CNN с Resnet-101, SSD с MobileNet-1, SSD с MobileNet-2. При обучении и обработке на изображениях учитывались два класса объектов: «пешеход» и «автомобиль».

Сети Faster R-CNN показали преимущество в точности. Так, по результатам эксперимента самой высокой точностью обладает Faster R-CNN на базе сети InceptionResnet-2, но и среднее время обработки изображения для данного детектора самое большое. Архитектура SSD является наиболее подходящей для обработки изображений в режиме реального времени (особенно при использовании сетей MobileNet), но необходимо учитывать, что высокие требования к точности не всегда могут быть соблюдены. Нейросетевой детектор YOLOv3 обладает средними показателями точности и вычислительной эффективности по сравнению с другими исследованными детекторами.

Работа выполнена при финансовой поддержке стипендии Президента Российской Федерации молодым ученым и аспирантам (СП-2578.2018.5).

Литература

1. Лукьяница А.А., Шишкин А.Г. Цифровая обработка видеоизображений. – М.: Ай-Эс-Эс Пресс, 2009. – 518 с.

2. Алпатов Б.А., Бабаян П.В. Технологии обработки и распознавания изображений в бортовых системах технического зрения // Вестник Рязанского государственного радиотехнического университета. – Рязань. – 2017. – №2. – С. 34-44.

3. Алпатов Б.А., Бабаян П.В., Балашов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. – М.: Радиотехника, 2008. – 176 с.

4. Alpatov B.A., Babayan P.V., Ershov M.D. Vehicle Detection and Counting System for Real-Time Traffic Surveillance // Proceedings of 7th Mediterranean Conference on Embedded Computing (MECO). – IEEE, 2018. – pp. 120-123.

5. Gouk H.G.R., Blake A.M. Fast sliding window classification with convolutional neural networks // Proceedings of the 29th International Conference on Image and Vision Computing, New Zealand. – ACM, 2014. – pp. 114-118.

6. Boser B.E., Guyon I.M., Vapnik V.N. A training algorithm for optimal margin classifiers // Proceedings of the fifth annual workshop on Computational learning theory. – ACM, 1992. – pp. 144-152.

7. Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – pp. 779-788.

8. Redmon J., Farhadi A. YOLO9000: better, faster, stronger // arXiv preprint, arXiv:1612.08242. – 2016. – 9 p.

9. Redmon J., Farhadi A. YOLOv3: An incremental improvement // Tech report, arXiv:1804.02767. – 2018. – 6 p.

10. Bishop C.M. Pattern Recognition and Machine Learning. – Springer-Verlag, New York, 2006. – 738 p.

11. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // Extended tech report, arXiv:1506.01497. – 2016. – 14 p.

12. Girshick R.B., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2014. – 21 p.

13. Girshick R. Fast R-CNN // IEEE International Conference on Computer Vision (ICCV). – 2015. – 9 p.

14. Uijlings J.R.R., van de Sande K.E.A., Gevers T., Smeulders A.W.M. Selective Search for Object Recognition // International Journal of Computer Vision. – 2013. – Vol. 104. – pp. 154-171.

15. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu Ch.-Y., Berg A.C. SSD: Single Shot MultiBox Detector // European Conference on Computer Vision (ECCV), Springer, Cham. – 2016. – Vol. 9905. – pp. 21-37.

16. Wan S., Chen Z., Zhang T., Zhang B., Wong K. Bootstrapping Face Detection with Hard Negative Examples // arXiv:1608.02236. – 2016. – 7 p.

17. Geiger A., Lenz P., Urtasun R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite // Conference on Computer Vision and Pattern Recognition (CVPR). – 2012. – 8 p.

18. Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B. The Cityscapes Dataset for Semantic Urban Scene Understanding // Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – 11 p.