

Исследование эффективности популярных нейросетевых моделей детектирования объектов в поле наблюдения

С. М. Борзов*, Е. С. Нежевенко, О. И. Потатуркин

Институт автоматизации и электрометрии СО РАН, 630090, Новосибирск, Россия

*Контактный автор: Борзов Сергей Михайлович, e-mail: borzov@iae.nsk.su

Поступила 17 октября 2023 г., доработана 01 ноября 2023 г., принята в печать 07 ноября 2023 г.

Выполнен краткий обзор основных подходов к решению задач обнаружения, определения координат и классификации объектов по их изображениям с применением нейросетевых технологий. Рассмотрены наиболее распространенные методы YOLO и Faster RCNN, применяющие одно- и двухэтапные детекторы на основе предсказания положения объектов. Выполнено экспериментальное сравнение их эффективности на примере классификации изображений 3D-моделей автомобилей, схожих по общим очертаниям и габаритам. Отмечено, что одноэтапные методы позволяют осуществлять обработку видеопотоков в реальном времени, однако уступают по точности при разделении близких классов объектов сложной формы.

Ключевые слова: нейросетевые технологии, обработка изображений, детектирование объектов, сверточные нейронные сети, глубокое обучение.

Цитирование: Борзов С.М., Нежевенко Е.С., Потатуркин О.И. Исследование эффективности популярных нейросетевых моделей детектирования объектов в поле наблюдения. Вычислительные технологии. 2023; 28(6):68–80. DOI:10.25743/ICT.2023.28.6.007.

Введение

Для повышения эффективности современных информационно-управляющих систем, основанных на обработке данных видеонаблюдения, требуются разработка и развитие математических методов и программно-аппаратных средств обработки последовательностей регистрируемых изображений с целью обнаружения и определения координат объектов заданных классов. Опыт последних десятилетий показал, что наибольшие успехи в рассматриваемой области связаны с применением нейросетевых технологий, позволяющих в реальном времени анализировать не только отдельные цифровые изображения, но и видеопотоки [1–3]. Обнаружение объектов сводится к выделению прямоугольных областей, очерчивающих искомые объекты на цифровом изображении. Классификация подразумевает отнесение объектов, содержащихся в выделенных областях, к одному из предварительно заданных классов.

При обнаружении и классификации объектов в поле наблюдения для достижения наилучшего результата необходимо осуществить полный перебор окон анализа (или ограничивающих их рамок) не только по их положению на изображении, но и по размеру, поскольку объекты могут быть разных масштабов и находиться в разных местах изображений. Это делает такой перебор неэффективным методом, занимающим очень

большое время. Для уменьшения количества рассматриваемых ограничивающих рамок выделяют два основных параллельно развивающихся подхода: двух- и одноэтапные.

Методы R-CNN [4], Fast R-CNN [5], Faster R-CNN [6] и т. п. (методы, основанные на регионах интереса) включают два этапа. На первом этапе селективным поиском или с помощью специального слоя нейронной сети выделяются регионы интереса — области, с высокой вероятностью содержащие внутри себя объекты. На втором этапе выбранные регионы подаются на вход базовой сверточной нейронной сети, преобразуются и затем рассматриваются классификатором для определения принадлежности к конкретным классам и регрессором, уточняющим местоположение ограничивающих рамок. Одноэтапные методы YOLO [7], SSD [8] и т. п. не используют отдельный алгоритм для генерации регионов, вместо этого осуществляется предсказание координат определенного количества ограничивающих рамок с различными характеристиками, такими, как результаты классификации и степень уверенности. В дальнейшем выполняется коррекция местоположения рамок.

Как можно видеть, разделение подходов достаточно условно, в обоих случаях осуществляются выделение подозрительных фрагментов изображений и их детальный анализ. Только при реализации первого подхода задачи его выделения и классификации решают отдельные методы (модели, сети), а при реализации второго эти шаги предусмотрены (“спрятаны”) в одном методе (в модели, в сети).

В настоящее время опубликованы объемные и обстоятельные обзоры методов обнаружения объектов. Несомненный интерес представляет большой обзор статей за 20 лет [1], в котором, в частности, приведена диаграмма увеличения числа публикаций с 1998 по 2018 г. по данной теме в Google Scholar. Следует отметить тематический обзор [2], в котором рассмотрены критерии качества алгоритмов обнаружения, классические статистические обнаружители, использующие математические модели случайных полей, а также современные решения на базе архитектур сверточных нейронных сетей из популярного фреймворка для машинного обучения TensorFlow. Приведен взгляд авторов на основные перспективы и тенденции в задаче обнаружения объектов на изображениях.

В [9] рассматриваются задачи поиска и классификации объектов на изображениях с использованием методов семейства R-CNN. Приведены результаты тестирования нейросетевых моделей с различными архитектурами на наборах данных VOC2007 и COCO. Проанализированы результаты исследования точности распознавания в зависимости от различных гиперпараметров обучения. Исследовано изменение значения времени определения местоположения объекта в зависимости от различных архитектур нейронной сети. Показано существенное превосходство модели Faster R-CNN из-за использования в качестве алгоритма генерации областей интереса специальной сети формирования соответствующих предложений.

В [10] выполнено тщательное сравнение производительности различных моделей обнаружения объектов YOLO семейства моделей YOLOv5, YOLOv6 и YOLOv7, которое дает полное и всестороннее представление о том, какую модель стоит использовать в той или иной задаче с точки зрения ее сильных сторон, недостатков и много другого. В частности, показано, что самыми быстрыми моделями на CPU являются модели YOLOv5 Nano и YOLOv5 Nano P6. Они обеспечивают скорость чуть выше 30 кадров в секунду. Для достижения большей скорости обработки необходимо использовать графический процессор. Так, для моделей YOLOv6 с новейшими графическими процессорами NVIDIA RTX 4090 и Tesla V100 быстродействие составляет от 150 до 170 кадров в секунду.

В работе [11] выполнено сравнение указанных моделей на примере решения задач обнаружения изображений людей. Показано преимущество Faster R-CNN в точности, а YOLOv5 в быстродействии (при вполне удовлетворительной, как отмечают авторы, точности).

В работе [12] рассматриваемые модели сравниваются на примере решения задачи классификации марок автомобилей. В частности, сделан вывод о безоговорочном преимуществе YOLO над R-CNN как по точности, так и по быстродействию. Но в этой работе в качестве выделяемых классов использовались автомобили разных категорий — микроавтобус, грузовик и т. п. Эти классы могут быть отличимы просто по габаритным размерам или по общим очертаниям кузова.

В работе [13] приведен углубленный анализ методов Faster R-CNN, YOLO, подробно описаны соответствующие архитектуры и их предлагаемые варианты, перечислены ограничения существующих работ и не изученные аспекты этой темы исследования. Отмечается необходимость развития методов данного класса, предлагается осуществлять это на основе YOLOv5. Результаты сравнения эффективности моделей на практике не приводятся.

Целью настоящей работы является экспериментальное сравнение эффективности указанных нейросетевых методов детектирования объектов. При этом для нашего исследования принципиально важна проверка эффективности нейросетевых моделей при классификации сложных изображений близких классов, когда объекты различаются лишь отдельными элементами и весьма схожи по общим очертаниям и габаритам. В качестве тестовых классов выбраны три модели автомобиля, удовлетворяющие данному требованию.

1. Методы обработки

Один из наиболее распространенных двухэтапных методов — Region-CNN (R-CNN, region-based convolutional neural network). Он применяет алгоритм, основанный на сверточных нейронных сетях и реализуемый в три шага.

Вместо того чтобы использовать для поиска изображений скользящие окна фиксированного размера, на первом шаге алгоритм пытается найти селективным поиском “регионы” — прямоугольные рамки разных размеров, которые предположительно содержат объекты. Это обеспечивает более быстрое и эффективное нахождение объектов независимо от их размеров, дальности, угла зрения. Суммарное количество регионов для каждого изображения, сгенерированных на первом шаге, примерно равно двум тысячам.

На втором шаге сверточная нейронная сеть формирует вектор-признак фиксированной длины для каждого региона. Предварительно найденные регионы при помощи аффинных преобразований приводятся к размеру, который нужно подать на вход сверточной нейронной сети. Также вместо аффинных преобразований можно использовать различные варианты паддинга (добавление “ложных” пикселей до требуемого размера фрагмента) либо расширять ограничивающие рамки до необходимых размеров. В качестве сверточной нейронной сети зачастую используется архитектура CaffeNet [14], извлекающая для каждого региона порядка 4096 признаков.

На последнем этапе вектор-признаки регионов обрабатываются набором линейных классификаторов на основе метода опорных векторов (support vector machine — SVM) [15, 16], осуществляющих классификацию объектов.

Метод Fast R-CNN [5], в отличие от оригинального алгоритма R-CNN, использует единоразовую свертку перед нахождением регионов селективным поиском. За счет этого время обработки одного тестового изображения снизилось более чем на порядок — до нескольких секунд. Метод Faster R-CNN [6], призванный еще более сократить время обработки, определяет регионы с помощью отдельного модуля region proposal network (RPN). RPN является сверточной сетью, выполняющей роль генератора регионов по общей карте признаков исходного изображения, сформированной базовой CNN. Далее с регионами Faster R-CNN работает аналогично Fast R-CNN. Для каждого региона рассчитывается низкоразмерный признак (размерностью 256 или 512), который подается на два однотипных полносвязных слоя, осуществляющих собственно классификацию и уточнение положения ограничивающих рамок.

Первой попыткой сделать возможным обнаружение объектов в поле наблюдения в реальном времени был метод YOLO (you look only once), предложенный в 2016 г. В рамках метода YOLO исходные изображения сначала разбиваются на сетку из $N \times N$ ячеек. Если центр объекта попадает внутрь координат ячейки, то эта ячейка считается ответственной за определение параметров местонахождения объекта. Для каждой ячейки рассматривается несколько вариантов местоположения ограничивающих рамок для одного и того же объекта. Каждый из этих вариантов характеризуется пятью значениями — координатами центра ограничивающей рамки, ее шириной и высотой, а также достоверностью обнаружения — степенью уверенности сети в том, что ограничивающая рамка содержит в себе какой-либо объект. Также для каждого класса объектов и каждой ячейки определяется достоверность классификации — степень уверенности сети в том, что данная ячейка содержит в себе объект данного класса.

Таким образом, обнаружение объектов рассматривается как единая проблема регрессии, прямо от пикселей изображения к координатам ограничительной рамки и достоверностям обнаружения и классификации. При использовании метода YOLO, чтобы предсказать, какие объекты присутствуют и где они находятся, изображение просматривается только один раз. Последний слой сети, принимающий конечное решение о положении на изображении и классе объектов, на своем выходе формирует тензор размерности $N \times N \times (5B + C)$, где B — количество предсказываемых ограничивающих рамок для ячейки, C — заданное количество классов объектов.

Улучшенная версия модели YOLOv2 [17] отличается от предшественницы использованием батчевой (пакетной) нормализации на сверточных слоях, обучением модели на изображениях с повышенным разрешением, использованием ключевых рамок для предсказания местонахождения объектов, применением для кластеризации алгоритма k -средних. YOLOv3 [18], в свою очередь, является небольшим улучшением YOLOv2. В этой версии используется логистическая регрессия для оценок достоверностей ограничивающих рамок вместо суммы квадратов ошибок для условий классификации в YOLO и YOLOv2; используются несколько независимых логистических классификаторов для каждого класса; добавлены межуровневые соединения между уровнями прогнозирования ограничивающих рамок; используются архитектуры DarkNet и ResNet для сверточных сетей.

Следующие версии YOLO [19–23] получили ряд дальнейших усовершенствований, они в первую очередь ориентированы на наиболее эффективное использование вычислителей с массивным параллелизмом, таких как GPU (graphics processing unit) и NPU (neural processing unit).

2. Экспериментальные исследования

Для сравнения указанных методов в рамках настоящей работы выполнено исследование эффективности нейронных сетей Faster RCNN на базе ResNet50 [24] и YOLOv5 путем анализа результатов классификации с их помощью изображений, синтезированных с применением цифровых 3D-моделей автомобилей. Использовались предобученные на изображениях баз ImageNet [25] и COCO [26] сети из библиотеки PyTorch, которые дополнительно обучались на изображениях интересующих нас классов объектов с применением специальным образом сформированных обучающих выборок (ОВ).

При синтезе ОВ использовались цифровые 3D-модели техники трех классов (объект 1, объект 2 и объект 3) (рис. 1). При генерации проекций применялось вращение модели вокруг вертикальной оси с постоянным шагом по углу $360^\circ/32$ при одних и тех же условиях освещения. Это позволило создать представительную для данного вычислительного эксперимента обучающую выборку.

Поскольку на практике нас интересуют обнаружение и классификация объектов на максимально возможной дальности, размер их изображений уменьшался примерно до 100 пикселей по максимальной координате. Для типичного размера автомобиля 4 м при угле зрения системы наблюдения 12° и разрешении 1920×1080 (Full HD) это приблизительно соответствует дальности 400 м. Далее, для удобства работы, изображения объектов в различных ракурсах были собраны в коллажи. Пример такого изображения обучающей выборки с 25 различными ракурсами объекта 1 приведен на рис. 2.

Обучение сетей выполнялось на шести подобных коллажах с изображениями указанных объектов в 31 различном ракурсе для каждого класса с применением алгоритма обратного распространения ошибки (backpropagation). Сначала в предобученных сетях Faster RCNN и YOLO из библиотеки PyTorch производилось изменение последних полносвязных слоев, выполняющих функцию классификатора таким образом, чтобы размерность выходного тензора сетей соответствовала количеству классов. Затем осуществлялся собственно алгоритм обучения. В основе идеи алгоритма лежит использование выходной ошибки нейронной сети для вычисления величин коррекции весов нейронов в ее скрытых слоях. В процессе обучения веса нейронов сети корректируются после подачи на ее вход каждого обучающего примера. На каждой итерации происходит два прохода сети — прямой и обратный. На прямом входной вектор распространяется от входов сети к ее выходам и формирует некоторый выходной вектор, соответствующий текущему состоянию весов. Затем вычисляется ошибка нейронной сети как разность между фактическим и целевым значениями. На обратном проходе эта ошибка распространяется от выхода сети к ее входам и производится коррекция весов нейронов таким образом, чтобы уменьшить значение ошибки сети. При этом настройка весов



Рис. 1. Классифицируемые модели техники: *a* — объект 1 (Lada Priora); *б* — объект 2 (Toyota Corolla); *в* — объект 3 (Hyundai Genesis)

Fig. 1. Classified objects: *a* — object 1 (Lada Priora); *б* — object 2 (Toyota Corolla); *в* — object 3 (Hyundai Genesis)

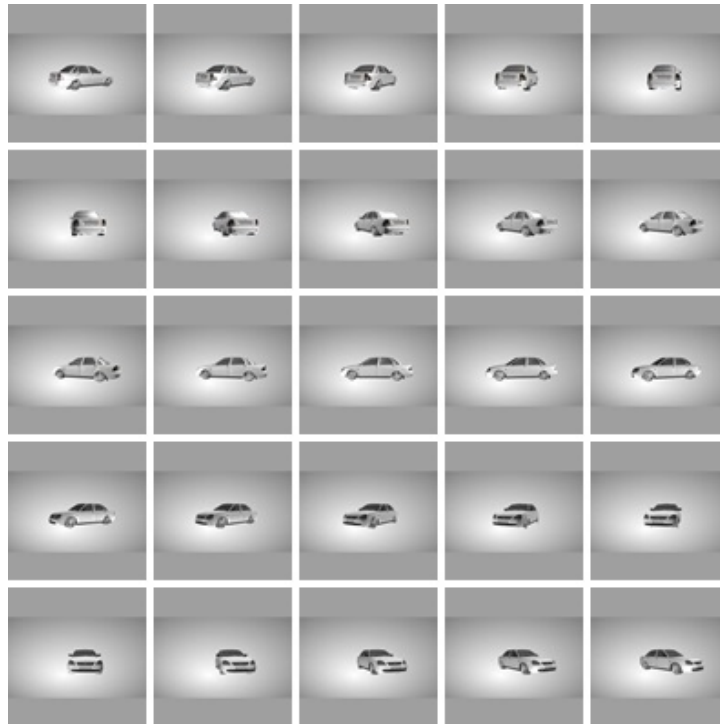


Рис. 2. Пример изображения обучающей выборки

Fig. 2. Sample image from the training set

может осуществляться только у части слоев, остальные сохраняются без изменений (замораживаются).

Обучение сети YOLO выполнялось тремя способами:

- всех слоев без их замораживания;
- с замораживанием десяти слоев базовой CNN-сети;
- с замораживанием всех слоев, кроме последнего.

Обучение сети Faster RCNN выполнялось двумя способами: всех слоев без их замораживания и с замораживанием всех слоев, кроме последнего.

После обучения выполнена проверка точности обнаружения и классификации объектов на изображениях ОБ (при предъявлении на вход сети тех же коллажей изображений, на которых осуществлялось обучение) для каждого из перечисленных выше случаев. Полученные результаты представлены в табл. 1–5.

Установлено, что для модели YOLO лучшие результаты достигаются при обучении с десятью замороженными слоями. В этом случае верно классифицированы 63 объекта из 93 (табл. 2). При обучении всех слоев и только последнего слоя правильные решения приняты для 52 и 54 объектов (соответственно табл. 1 и 3).

Для модели Faster RCNN на базе ResNet50 лучшие результаты достигаются при обучении всех слоев без их замораживания. Все объекты ОБ в этом случае были обнаружены и классифицированы верно (табл. 4). При обучении с замораживанием всех слоев сети Faster RCNN, кроме последнего, результаты получились близкие, но несколько хуже (табл. 5). Отметим, что обнаружение объектов ОБ обеими моделями для всех рассматриваемых способов обучения выполнено без ошибок. Время обработки одного изображения размером 1024×1024 пиксела на персональном компьютере с процессором Intel® Core™ i3-8100 CPU 3.60 ГГц и оперативной памятью 16 ГБ (без использования GPU) для метода Faster RCNN составило 2.5 с, для YOLOv5 — 0.5 с.

Т а б л и ц а 1. Обнаружение и классификация изображений ОВ при обучении всех слоев YOLO

Table 1. Detection and classification of training set images when training all layers of the YOLO

Файл ОВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	31	0	0	31
Коллаж объекта 2	31	0	21	10
Коллаж объекта 3	31	0	0	31

Всего верно классифицированы 52 из 93 объектов ОВ.

Т а б л и ц а 2. Обнаружение и классификация изображений ОВ при обучении с замораживанием десяти слоев базовой сети YOLO

Table 2. Detection and classification of images of the training set during training with freezing of 10 layers of the YOLO base network

Файл ОВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	31	10	14	7
Коллаж объекта 2	31	0	30	1
Коллаж объекта 3	31	3	1	27

Всего верно классифицированы 67 из 93 объектов ОВ.

Т а б л и ц а 3. Обнаружение и классификация изображений ОВ при обучении с замораживанием всех слоев YOLO, кроме последнего

Table 3. Detection and classification of training set images when training all layers of the YOLO except the last one

Файл ОВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	31	10	14	7
Коллаж объекта 2	31	0	30	1
Коллаж объекта 3	31	3	1	27

Всего верно классифицированы 54 из 93 объектов ОВ.

Т а б л и ц а 4. Обнаружение и классификация изображений ОВ при обучении всех слоев Faster RCNN

Table 4. Detection and classification of training set images when training all layers of the Faster RCNN

Файл ОВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	31	31	0	0
Коллаж объекта 2	31	0	31	0
Коллаж объекта 3	31	0	0	31

Всего верно классифицированы 93 из 93 объектов ОВ.

Т а б л и ц а 5. Обнаружение и классификация изображений ОВ при обучении всех слоев Faster RCNN, кроме последнего

Table 5. Detection and classification of training set images when training all layers of the Faster RCNN except the last one

Файл ОВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	31	30	0	1
Коллаж объекта 2	31	0	31	0
Коллаж объекта 3	31	0	0	31

Всего верно классифицированы 92 из 93 объектов ОВ.

Т а б л и ц а 6. Обнаружение и классификация изображений ТВ при обучении всех слоев сети YOLO

Table 6. Detection and classification of test set images when training all layers of the YOLO

Файл ТВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	25	0	2	23
Коллаж объекта 2	25	1	9	15
Коллаж объекта 3	25	0	11	14

Всего верно классифицированы 23 из 75 объектов ТВ.

Т а б л и ц а 7. Обнаружение и классификация изображений ТВ при обучении с замораживанием десяти слоев базовой сети YOLO

Table 7. Detection and classification of images of the test set during training with freezing of 10 layers of the YOLO base network

Файл ТВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	24	0	18	6
Коллаж объекта 2	24	1	23	1
Коллаж объекта 3	25	1	24	0

Всего верно классифицированы 23 из 75 объектов ТВ.

Т а б л и ц а 8. Обнаружение и классификация изображений ТВ при обучении с замораживанием всех слоев сети YOLO, кроме последнего

Table 8. Detection and classification of test set images when training all layers of the YOLO except the last one

Файл ТВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	13	1	11	1
Коллаж объекта 2	16	0	16	0
Коллаж объекта 3	24	0	24	0

Всего верно классифицированы 17 из 75 объектов ТВ.

Т а б л и ц а 9. Обнаружение и классификация изображений ТВ при обучении всех слоев сети Faster RCNN

Table 9. Detection and classification of test set images when training all layers of the Faster RCNN

Файл ТВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	25	18	0	7
Коллаж объекта 2	25	9	10	6
Коллаж объекта 3	25	2	0	23

Всего верно классифицированы 51 из 75 объектов ТВ.

Т а б л и ц а 10. Обнаружение и классификация изображений ТВ при обучении всех слоев сети Faster RCNN, кроме последнего

Table 10. Detection and classification of test set images when training all layers of the Faster RCNN except the last one

Файл ТВ	Обнаружение	Объект 1	Объект 2	Объект 3
Коллаж объекта 1	25	26	0	0
Коллаж объекта 2	25	7	16	2
Коллаж объекта 3	25	21	1	3

Всего верно классифицированы 44 из 75 объектов ТВ.

Далее сравнение эффективности сетей проводилось на примере обнаружения объектов на коллажах изображений тестовой выборки (ТВ), сгенерированных с применением тех же трехмерных моделей техники в близких, но других 25 ракурсах каждого объекта.

Результаты обнаружения и классификации объектов ТВ с применением модели YOLO приведены в табл. 6–8. Видно, что эта модель плохо справилась с решением задачи, особенно с классификацией объектов. Результаты для Faster RCNN приведены в табл. 9 и 10. Они существенно лучше. Наиболее успешно классификация выполнена при обучении всех слоев модели (без их замораживания). В этом случае верно классифицирован 51 объект тестовой выборки из 75.

Заключение

Рассмотрена задача обнаружения и классификации объектов на изображениях с применением моделей нейронных сетей Faster R-CNN на базе ResNet50 и YOLOv5. Использовались предобученные на изображениях баз ImageNet и COCO сети из библиотеки PyTorch, которые дополнительно обучались на изображениях 3D-моделей автомобилей с применением специальным образом сформированных обучающих выборок. В ходе анализа результатов классификации схожих по общим очертаниям и габаритам автомобилей установлено, что модель Faster RCNN верно классифицирует 51 из 75 изображений объектов, тогда как YOLO — только 23. Таким образом, показано, что точность классификации для Faster RCNN существенно выше. Алгоритм YOLO работает быстрее алгоритмов семейства R-CNN за счет того, что поддерживает дробление на постоянное количество ячеек вместо того, чтобы определять регионы интереса и рассчитывать решение для каждого региона отдельно, однако качество распознавания объектов сложной формы у такого подхода существенно ниже. Это связано с ограниченным числом ограничивающих рамок для каждого потенциального объекта.

Благодарности. Работа выполнена при поддержке Министерства науки и высшего образования в рамках выполнения работ по Государственному заданию № 121022000116-0 для ИАиЭ СО РАН.

Список литературы

- [1] Zou Z., Shi Z. Guo Y., Ye J. Object detection in 20 years: a survey. *Proceedings of the IEEE*. 2023; 111(3):257–276.
- [2] Андриянов Н.А., Дементьев В.Е., Ташлинский А.Г. Обнаружение объектов на изображении: от критериев Байеса и Неймана–Пирсона к детекторам на базе нейронных сетей EfficientDet. *Компьютерная оптика*. 2022; 46(1):139–159. DOI:10.18287/2412-6179-СО-922.
- [3] Борзов С.М., Нежевенко Е.С. Нейросетевые технологии в задачах обнаружения и классификации объектов. *Автометрия*. 2023; 59(3):52–71. DOI: 10.15372/AUT20230307.
- [4] Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014: 580–587. Available at: <https://arxiv.org/abs/1311.2524>.
- [5] Girshick R. Fast RCNN. *Proceedings of the IEEE International Conference on Computer Vision*. 2015: 1440–1448. Available at: <https://arxiv.org/pdf/1504.08083.pdf>.

- [6] **Ren S., He K., Girshick R., Sun J.** Faster RCNN. Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*. 2015: 14. Available at: <https://arxiv.org/pdf/1506.01497.pdf>.
- [7] **Redmon J., Divvala S., Girshick R., Farhadi A.** You only look once: unified, realtime object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016: 779–788. Available at: <https://ieeexplore.ieee.org/document/7780460>.
- [8] **Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C.** SSD: single shot multibox detector. *Proceedings of the European Conference on Computer Vision*. Gewerbestrasse, Switzerland, Oct. 8–16, 2016: 21–37.
- [9] **Сирота А.А., Митрофанова Е.Ю., Милованова А.И.** Анализ алгоритмов поиска объектов на изображениях с использованием различных модификаций сверточных нейронных сетей. *Вестник ВГУ. Серия: Системный анализ и информационные технологии*. 2019; (3):123–137.
- [10] Performance comparison of YOLO object detection models — an intensive study. Available at: <https://learnopencv.com/performance-comparison-of-yolo-models>.
- [11] **Тимошкин М.С., Миронов А.Н., Леонтьев А.С.** Сравнение YOLO V5 и Faster R-CNN для обнаружения людей на изображении в потоковом режиме. *Международный научно-исследовательский журнал*. 2022; 6(120). DOI:10.23670/IRJ.2022.120.6.020. Адрес доступа: <https://research-journal.org/archive/6-120-2022-june/sravnenie-yolov5-i-fasterr-cnn-dlyaobnaruzheniya-lyudej-na-izobrazhenii-v-potokovom-rezhime> (дата обращения 31.10.2023).
- [12] **Kim J., Sung J.-Y., Park S.** Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE; 2020: 1–4.
- [13] **Maity M., Banerjee S., Chaudhuri S.S.** Faster R-CNN and YOLO based vehicle detection: a survey. *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE; 2021: 1442–1447.
- [14] **Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T.** Caffe: convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM International Conference on Multimedia*. 2014: 675–678. Available at: <https://arxiv.org/pdf/1408.5093.pdf>.
- [15] **Cristianini N.** An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press; 2000: 180.
- [16] **Chang C.C., Lin C.J.** LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2011; 2(3): Art. No 27. DOI:10.1145/1961189.1961199. Available at: <https://pdf4pro.com/view/libsvm-a-library-for-support-vector-machines-448847.html>.
- [17] **Redmon J., Farhadi A.** YOLO9000: better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017: 7263–7271. Available at: <https://arxiv.org/abs/1612.08242>.
- [18] **Redmon J., Farhadi A.** YOLOv3: an incremental improvement. Available at: <https://arxiv.org/pdf/1804.02767.pdf> (accessed 09.09.2023).
- [19] YOLOv5 GitHub. Available at: <https://github.com/ultralytics/yolov5> (accessed 14.09.2023).
- [20] YOLOv6 GitHub. Available at: <https://github.com/meituan/YOLOv6> (accessed 14.09.2023).

- [21] YOLOv7 GitHub. Available at: <https://github.com/WongKinYiu/yolov7> (accessed 14.09.2023).
- [22] Glenn J. Ultralytics YOLOv8. Available at: <https://github.com/ultralytics/ultralytics> (accessed 14.09.2023).
- [23] Li C., Li L., Geng Y., Jiang H., Cheng M., Zhang B., Ke Z., Xu X., Chu X. YOLOv6 v3.0: a full-scale reloading. Computer Vision and Pattern Recognition. Available at: <https://arxiv.org/abs/2301.05586> (accessed 14.09.2023).
- [24] Veit A., Wilber M.J., Belongie S.J. Residual networks are exponential ensembles of relatively shallow networks. AtXiv.org. 2016. DOI:10.48550/arXiv.1605.06431. Available at: <https://arxiv.org/pdf/1605.06431.pdf>.
- [25] Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L. ImageNet: a large-scale hierarchical image database. IEEE Computer Vision and Pattern Recognition (CVPR). 2009: 248–255. Available at: https://image-net.org/static_files/papers/imagenet_cvpr09.pdf.
- [26] COCO. Available at: <https://cocodataset.org/#home> (accessed 14.09.2023).

Вычислительные технологии, 2023, том 28, № 6, с. 68–80. © ФИЦ ИВТ, 2023
Computational Technologies, 2023, vol. 28, no. 6, pp. 68–80. © FRC ICT, 2023

ISSN 1560-7534
eISSN 2313-691X

INFORMATION TECHNOLOGIES

DOI:10.25743/ICT.2023.28.6.007

Investigation of the effectiveness of popular neural network models for detecting objects in the observation field

S. M. BORZOV*, E. S. NEZHEVENKO, O. I. POTATURKIN

Institute of Automation and Electrometry SB RAS, 630090, Novosibirsk, Russia

*Corresponding author: Sergey M. Borzov, e-mail: borzov@iae.nsk.su

Received October 17, 2023, revised November 01, 2023, accepted November 07, 2023.

Abstract

A brief overview of the main approaches used for solving the problems of detection, determination of coordinates and classification of objects by their images is performed using neural network technologies. The most common YOLO and Faster RCNN methods based on predicting the position of objects are considered using one- and two-stage detectors. At the first stage of the two-stage methods, we identify areas with a high probability containing objects inside themselves by selective search or using a special neural network. At the second stage they are examined by the classifier to determine whether or not they belong to specified classes, as well as to clarify their location and size. One-step methods do not use a separate algorithm to generate regions of interest. Instead, they simultaneously form a certain number of bounding boxes with different parameters in one pass and predict the class of the object. An experimental comparison of the effectiveness of these approaches has been performed. It is noted that single-stage methods allow processing video streams in real time, but they are inferior in accuracy when separating close classes of objects of complex shape.

Keywords: neural network technologies, image processing, object detection, convolutional neural networks, deep learning.

Citation: Borzov S.M., Nezhevenko E.S., Potaturkin O.I. Investigation of the effectiveness of popular neural network models for detecting objects in the observation field. Computational Technologies. 2023; 28(6):68–80. DOI:10.25743/ICT.2023.28.6.007. (In Russ.)

Acknowledgements. The work was carried out with the support of the Ministry of Science and Higher Education as part of the work on the State Task No. 121022000116-0 for the IA&E SB RAS.

References

1. **Zou Z., Shi Z. Guo Y., Ye J.** Object detection in 20 years: a survey. *Proceedings of the IEEE*. 2023; 111(3):257–276.
2. **Andriyanov N.A., Dementiev V.E., Tashlinskiy A.G.** Detection of objects in the images: from likelihood relationships towards scalable and efficient neural networks. *Computer Optics*. 2022; 46(1):139–159. DOI:10.18287/2412-6179-CO-922. (In Russ.)
3. **Borzov S.M., Nezhevenko E.S.** Neural network technologies for detection and classification of objects. *Optoelectronics, Instrumentation and Data Processing*. 2023. 59(3):329–345. (In Russ.)
4. **Girshick R., Donahue J., Darrell T., Malik J.** Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014: 580–587. Available at: <https://arxiv.org/abs/1311.2524>.
5. **Girshick R.** Fast RCNN. *Proceedings of the IEEE International Conference on Computer Vision*. 2015: 1440–1448. Available at: <https://arxiv.org/pdf/1504.08083.pdf>.
6. **Ren S., He K., Girshick R., Sun J.** Faster RCNN. Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*. 2015: 14. Available at: <https://arxiv.org/pdf/1506.01497.pdf>.
7. **Redmon J., Divvala S., Girshick R., Farhadi A.** You only look once: unified, realtime object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016: 779–788. Available at: <https://ieeexplore.ieee.org/document/7780460>.
8. **Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C.** SSD: single shot multibox detector. *Proceedings of the European Conference on Computer Vision*. Gewerbestrasse, Switzerland, Oct. 8–16, 2016: 21–37.
9. **Sirota A.A., Mitrofanova E.Yu., Milovanova A.I.** Analysis of algorithms for searching objects in images using various modifications of convolutional neural network. *Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies*. 2019; (3):123–137. (In Russ.)
10. Performance comparison of YOLO object detection models — an intensive study. Available at: <https://learnopencv.com/performance-comparison-of-yolo-models>.
11. **Timoshkin M.S., Mironov A.N., Leontev A.S.** Comparison of YOLOv5 and Faster R-CNN for detecting people in the image in streaming mode. *International Scientific Research Journal*. 2022; 6(120). DOI:10.23670/IRJ.2022.120.6.020. Available at: <https://research-journal.org/archive/6-120-2022-june/sravnenie-yolov5-i-fasterr-cnn-dlyaobnaruzheniya-lyudej-na-izobrazhenii-v-potokovom-rezhime>. (In Russ.)
12. **Kim J., Sung J.-Y., Park S.** Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE; 2020: 1–4.
13. **Maity M., Banerjee S., Chaudhuri S.S.** Faster R-CNN and YOLO based vehicle detection: a survey. *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE; 2021: 1442–1447.
14. **Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T.** Caffe: convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM International Conference on Multimedia*. 2014: 675–678. Available at: <https://arxiv.org/pdf/1408.5093.pdf>.
15. **Cristianini N.** An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press; 2000: 180.
16. **Chang C.C., Lin C.J.** LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2011; 2(3). 27. Available at: <https://pdf4pro.com/view/libsvm-a-library-for-support-vector-machines-448847.html>.
17. **Redmon J., Farhadi A.** YOLO9000: better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017: 7263–7271. Available at: <https://arxiv.org/abs/1612.08242>.
18. **Redmon J., Farhadi A.** YOLOv3: an incremental improvement. Available at: <https://arxiv.org/pdf/1804.02767.pdf> (accessed 09.09.2023).
19. YOLOv5 GitHub. Available at: <https://github.com/ultralytics/yolov5> (accessed 14.09.2023).

20. YOLOv6 GitHub. Available at: <https://github.com/meituan/YOLOv6> (accessed 14.09.2023).
21. YOLOv7 GitHub. Available at: <https://github.com/WongKinYiu/yolov7> (accessed 14.09.2023).
22. **Glenn J.** Ultralytics YOLOv8. Available at: <https://github.com/ultralytics/ultralytics> (accessed 14.09.2023).
23. **Li C., Li L., Geng Y., Jiang H., Cheng M., Zhang B., Ke Z., Xu X., Chu X.** YOLOv6 v3.0: a full-scale reloading. *Computer Vision and Pattern Recognition*. Available at: <https://arxiv.org/abs/2301.05586> (accessed 14.09.2023).
24. **Veit A., Wilber M.J., Belongie S.J.** Residual networks are exponential ensembles of relatively shallow networks. *AtXiv.org*. 2016. DOI:10.48550/arXiv.1605.06431. Available at: <https://arxiv.org/pdf/1605.06431.pdf>.
25. **Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L.** ImageNet: a large-scale hierarchical image database. *IEEE Computer Vision and Pattern Recognition (CVPR)*. 2009: 248–255. Available at: https://image-net.org/static_files/papers/imagenet_cvpr09.pdf.
26. COCO. Available at: <https://cocodataset.org/#home> (accessed 14.09.2023).