



Math-Net.Ru

All Russian mathematical portal

T. S. Katermina, E. V. Lazorenko, Artificial intelligence elements for the task of determining the position of the vehicle in the image, *Comp. nanotechnol.*, 2022, Volume 9, Issue 3, 9–18

DOI: 10.33693/2313-223X-2022-9-3-9-18

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use

<http://www.mathnet.ru/eng/agreement>

Download details:

IP: 18.97.14.80

February 6, 2025, 21:56:54



ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

05.13.18

*МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ,
ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ ПРОГРАММ
MATHEMATICAL MODELING, NUMERICAL METHODS
AND COMPLEX PROGRAMS*

1.2.1

*ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING*

DOI: 10.33693/2313-223X-2022-9-3-9-18

Элементы искусственного интеллекта для задачи определения положения автотранспортного средства на изображении

Т.С. Катермина^а ©, Е.В. Лазоренко^б ©

Нижевартовский государственный университет,
г. Нижневартовск, Ханты-Мансийский автономный округ – Югра,
Российская Федерация

^а E-mail: nggu-lib@mail.ru

^б E-mail: rolaraltis@hotmail.com

Аннотация. Статья посвящена решению задачи определения границ автотранспортного средства на изображении, как промежуточного этапа для решения других, более локальных задач, связанных с идентификацией автотранспорта на изображении или в видео потоке. В статье подробно рассматриваются существующие методы и подходы к решению задач компьютерного зрения, в том числе современные архитектуры нейронных сетей. В качестве основной модели была выбрана сверточная нейронная сеть Tiny-YOLO-InceptionResNet, которая была модифицирована в процессе проведения исследования. Архитектура полученной нейронной сети приведена в данной работе. Перед обучением нейронной сети производилась работа по подготовке и предобработке набора данных, которая позволила более рационально использовать вычислительные ресурсы во время обучения. В результате проведенного исследования была разработана модель нахождения границ автотранспортного средства на изображении, точность которой равна 88%.

Ключевые слова: компьютерное зрение, нейронные сети, сверточные нейронные сети, распознавание изображений, YOLO, искусственный интеллект

ССЫЛКА НА СТАТЬЮ: Катермина Т.С., Лазоренко Е.В. Элементы искусственного интеллекта для задачи определения положения автотранспортного средства на изображении // Computational nanotechnology. 2022. Т. 9. № 3. С. 9–18.
DOI: 10.33693/2313-223X-2022-9-3-9-18

Artificial Intelligence Elements for the Task of Determining the Position of the Vehicle in the Image

T.S. Katermina^a ©, E.V. Lazorenko^b ©

Nizhnevartovsk State University,
Nizhnevartovsk, Khanty-Mansi Autonomous Okrug – Yugra, Russian Federation

^a E-mail: nggu-lib@mail.ru

^b E-mail: rolalartis@hotmail.com

Abstract. The article is devoted to solving the problem of determining the boundaries of the vehicle on the image. This task is an intermediate step to solve other, more local tasks related to the identification of vehicles in the image or video stream. The article in detail considers existing methods and approaches to solving problems of computer vision, including modern architectures of neural networks. Tiny-YOLO-InceptionResNet was chosen as the primary neural network and was modified during the research process. The architecture of the resulting neural network is given in this paper. The training of the neural network was preceded by the preparation of a data set that allowed for a more rational use of computing resources during training. As a result of the research the model of finding the boundaries of the vehicle on the image was developed. The accuracy of this model is 88%.

Key words: computer vision, neural networks, convolutional neural networks, image recognition, YOLO, artificial intelligence

FOR CITATION: Katermina T.S., Lazorenko E.V. Artificial Intelligence Elements for the Task of Determining the Position of the Vehicle in the Image. *Computational Nanotechnology*. 2022. Vol. 9. No. 3. Pp. 9–18. (In Rus.) DOI: 10.33693/2313-223X-2022-9-3-9-18

ВВЕДЕНИЕ. ПОСТАНОВКА ПРОБЛЕМЫ

Задачи распознавания образов и обнаружения объектов являются частными задачами компьютерного зрения. Главная задача этой области заключается в сопоставлении визуальной информации с некоторыми предопределенными классами объектов. Необходимость решения такого рода задач возникает в самых различных областях, начиная от обработки простых цифровых изображений и заканчивая показаниями медицинских и военных приборов. В данной работе рассматривается вопрос построения нейронной сети для определения положения автотранспортного средства на изображении. Решение этой задачи является необходимым этапом для таких более локальных задач, как идентификация транспортного средства, обнаружение автомобиля-нарушителя, учет транспортных средств предприятия или автоматический расчет стоимости услуг роботизированных автомоек. Нейронная сеть, описываемая в данной статье, в дальнейшем будет использоваться как часть распознающей модели для программного комплекса роботизированной автомойки.

МЕТОДЫ РЕШЕНИЯ РАССМОТРЕННОЙ ЗАДАЧИ

Распознавание и обнаружение объектов осуществляются схожими методами, но реализуются по-разному.

В данной статье для решения поставленной задачи используются методы, основанные на нейросетевых подходах глубокого обучения, которые в последнее время приобрели значительное распространение. Модели глубокого обучения, такие как сверточные нейронные сети, используются для автоматического изучения признаков объекта идентификации. Рассмотрим этот и другие подходы и методы к решению задач компьютерного зрения.

РАСПОЗНАВАНИЕ ОБЪЕКТОВ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

Методы машинного обучения разнообразны и пользуются значительной популярностью при решении задачи распознавания образов. Распространенными примерами методов машинного обучения являются:

- извлечение функций HOG (Histogram of Oriented Gradients) с помощью модели машинного обучения SVM (support vector machine);
- алгоритм Виоля–Джонса, для распознавания различных объектов.

Гистограмма направленных градиентов (HOG)

Метод заключается в нахождении дескрипторов особых точек на основе обработки векторов направлений

Катермина Т.С., Лазоренко Е.В.

в локальных областях изображения [Rybiski et al., 2010]. Находится количество каждого из направлений.

Каждый объект, в том числе его форма, силуэт или общий вид, согласно идее данного подхода, описывается исходя из следующих характеристик: выделяются края объекта, для них находятся интенсивность и направление, для которых рассчитывается распределение градиентов. Таким образом, изображение объекта может быть представлено совокупностью этих распределений. Окрестность для нахождения градиентов определяется разделением изображения на связанные ячейки, небольшие области или клетки. Для каждой такой связанной области строится гистограмма направлений.

Дескриптор в таком случае представляет собой набор гистограмм, в котором все локальные гистограммы нормализованы. Для нормализации выбирается большой фрагмент изображения (блок), на нем вычисляется мера интенсивности. Используя для нормализации полученное значение, можно увеличить точность применения дескриптора. Такой вариант дескриптора НОГ инвариантен к освещенности, инвариантен к геометрическим и фотометрическим преобразованиям.

Метод опорных векторов (SVM)

Метод опорных векторов [Zehang Sun et al., 2002] основан на идее разделения гиперплоскостью данных в пространстве большей размерности, чем исходная. При этом гиперплоскость должна проходить таким образом, чтобы расстояние между разделяемыми векторами было максимально возможным. Чем лучше удастся построить разделение, тем больше будут отличаться друг от друга разделяемые данные, а значит средняя ошибка работы алгоритма будет минимальной.

По подобной идее основаны схожие алгоритмы, причем обучение проходит с учителем. Алгоритмы решают задачи нахождения регрессии и классификации и относятся к линейным алгоритмам.

Метод Виолы–Джонса (Viola–Jones object detection)

Алгоритм, представленный в 2001 г. Майклом Джонсом и Паулом Виолой [Xu et al., 2016], работает с образами объектов в видео потоке в реальном времени. Он используется, в первую очередь, для обнаружения и распознавания лиц людей, но способен также распознавать произвольные объекты на изображениях.

Алгоритм заключается в следующем: по изображению перемещается сканирующее окно, размер которого меньше, чем исходное изображение, шаг перемещения задается отдельно. К каждому полученному при помощи окна изображению применяется каскад слабых классификаторов, которые обучены для нахождения частей и черт лица. Найденные особенности лица отмечаются на изображении. В основе метода лежат следующие заключения:

- для ускорения вычислений черт и частей лица изображение преобразуется к интегральному виду и на нем производится поиск;

- для поиска объектов на изображении используются обученные классификаторы на основе признаков Хаара;
- в качестве метода подбора более релевантных признаков Хаара для искомой черты или части лица в данной области изображения применяется бустинг;
- результаты («истина» или «ложь») генерируются классификатором, которому на вход отправляются все полученные ранее признаки;
- окна, в которых не было найдено ни одной черты или части лица (например, детали одежды или фон), отбрасываются, при этом также используются обученные каскады признаков. Такой же принцип можно применять и для других произвольных объектов, предварительно обучив для них каскад классификаторов.

Таким образом, в методе используется следующий алгоритм: сканирующее окно перемещается по изображению и к каждому попавшему в окно фрагменту применяется классификатор. Вся последовательность действий при этом полностью автоматизирована, в том числе и в части обучения, и в части обнаружения максимально релевантных признаков. Система работает быстрее, чем в случае, если бы ее работа требовала участия человека.

В то же время, каскад классификаторов обучается достаточно медленно, но эта работа производится предварительно, а обученная модель очень быстро находит и идентифицирует черты и части лица человека, поэтому эта задача успешно решается данной группой методов.

Обнаружение объектов на основе функций Хаара

Характеристики, подобные Хаару, представляют собой идентификаторы внутри объекта, которые можно использовать для распознавания и обнаружения объектов [Atibi Mohamed et al., 2015].

Характеристики части изображения могут быть получены путем сравнения интенсивностей пикселей двух соседних прямоугольных областей и определения разницы между суммированием.

МЕТОДЫ ГЛУБОКОГО ОБУЧЕНИЯ

Ниже приведен краткий обзор моделей **сверточных нейронных сетей (CNN)** [Разинкин, Катермина, 2018] для обнаружения объектов. Сверточные нейронные сети широко применяются при распознавании образов различной природы, будь то объекты на изображении или, например, признаки заболеваний по набору анализов [Катермина, Сибегатулин, 2022] и т.д.

Региональные сверточные нейронные сети (RCNN) и их варианты

Это метод глубокого обучения для решения проблем обнаружения и сегментации объектов [Fan et al., 2016; Nguyen, 2019; Maity et al., 2021]. Он был

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

представлен в 2014 г. и представлял концепцию региональных сверточных нейронных сетей.

RCNN использует алгоритм выборочного поиска, чтобы предложить интересующую область на изображении, а после этого использует сверточную нейронную сеть для обнаружения присутствия интересующего объекта в предложенной области.

RCNN использует CNN для извлечения признаков и метод опорных векторов для классификации объектов.

RCNN может быть дорогостоящим с точки зрения вычислений, и для выполнения прогнозов на каждом изображении требуется в среднем 40–50 секунд. Его преемники, **Fast-RCNN** и **Faster-RCNN** [Shi et al., 2017; Manana et al., 2018] решают некоторые проблемы производительности и эффективности, характерные для RCNN. Кроме того, RCNN не может использоваться для приложений реального времени из-за времени, необходимого для выполнения и завершения прогнозов на изображениях.

YOLO

YOLO представляет собой одноэтапный процесс обнаружения объектов, модель нейронной сети, для которой требуется всего один проход изображения через сеть для обнаружения объекта [Lagosa et al., 2018; Miao, 2020]. Очевидным преимуществом является то, что YOLO предлагает большую эффективность по сравнению с RCNN.

YOLO работает путем наложения сеток на изображение, где каждая ячейка в сетке играет две роли:

- 1) предсказывает ограничивающие прямоугольники, и каждому ограничивающему прямоугольнику присваивается значение достоверности. Значение достоверности представляет собой представление вероятности того, что в ячейке содержится объект;
- 2) возвращает вероятности соотношения с некоторым классом.

Использование YOLO для обнаружения объектов имеет несколько преимуществ, главное из которых состоит в том, что его можно использовать в приложениях реального времени, таких как системы обнаружения объектов, встроенные в системы автономных транспортных средств.

При использовании методов глубокого обучения можно выделить два подхода к созданию модели, первый из которых заключается в проектировании и обучении сетевой архитектуры с нуля, включая структуру слоев и инициализацию значений весовых параметров. Второй подход заключается в использовании концепции **передачи обучения** и использовании предварительно обученной сети, которая обучена на большом наборе данных до начала специального обучения на пользовательских наборах данных.

Трансферное обучение [Shin et al., 2016] – это метод повторного использования знаний, полученных при решении проблемы, и применения полученных знаний к связанной, но отдельной проблеме.

**ДРУГИЕ МЕТОДЫ
РАСПОЗНАВАНИЯ ОБЪЕКТОВ**

Если к поставленной задаче можно применить другие, более простые методы и подходы, то можно обратить внимание на следующие:

- если существует небольшое изображение, при помощи которого можно произвести поиск совпадающих частей на изображении большего размера, то его можно использовать как шаблон, и метод в таком случае заключается в сопоставлении с шаблоном;
- в некоторых случаях можно провести анализ свойств изображения и объектов на нем, в том числе размера, цвета и формы объекта. Производится сегментация изображения, проводится анализ BLOB-объектов;
- если возможно использование цветных маркеров для объекта, то это можно сделать с большей эффективностью и меньшими затратами, чем при использовании методов машинного обучения.

Вообще, можно сформулировать основное правило компьютерного зрения – движение от простого к сложному. Сначала необходимо рассмотреть более простые и менее ресурсозатратные методы и подходы. Например, метод сегментации может оказаться более легко реализуемым, чем обучение каскадов Хаара. В таком случае решение может оказаться не только более дешевым, но и более надежным.

Если же задачу невозможно решить специальными методами, то можно выбрать более универсальный, для которого известны применяемые характеристики и который может применяться к обширному классу задач с известной ресурсозатратностью и эффективностью.

**ПРИМЕНЕНИЕ
ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ
К ЗАДАЧЕ РАСПОЗНАВАНИЯ АВТОТРАНСПОРТА****Выбор архитектуры нейронной сети YOLO**

Архитектура этой нейронной сети в первых блоках практически не отличается по «логике блоков» от других детекторов, то есть на вход подается картинка, дальше создаются featuremaps (карты признаков, карты характеристик) с помощью CNN (сверточной нейронной сети) (в YOLO используется своя CNN под названием Darknet-53), затем эти карты признаков определенным образом анализируются, выдавая на выходе позиции и размеры bounding boxes (границ объектов) и классы, к которым они принадлежат, и проходят через блок Sparse Prediction. Sparse Prediction определяет регионы и классифицирует их.

Тренировка YOLO происходит следующим образом: в модель подается изображение, происходит извлечение характеристик, полученный сверточный слой предсказывает все клетки, на которые была разделена картинка. И если этот слой предсказывает «неправду» о клетке на изображении, то получаем большой сигнал ошибки, который затем уменьшается при подаче в нейронную сеть следующих изображений.

Предобработка данных

Создание набора данных

Прежде чем подавать данные в нейронную сеть для обучения, необходимо подготовить набор данных. Для создания набора данных была использована онлайн-камера дорожного движения в городе Сочи. Данная видеокамера направлена на участок дороги с интенсивным движением. Данные с камеры обрабатываются следующим образом – с определенной периодичностью создаются моментальные снимки с видеопотока, которые затем проходят процесс разметки. В процессе тестирования и работы модели также использовалась информация с данной камеры.

Разметка данных

Разметка играет важную роль в обучении нейронной сети. Для поставленной задачи был разработан программный комплекс разметки изображений. В распознавании объектов из видео ряда стоит задача выделить нужный объект, отслеживать его и классифицировать. Результат работы данного программного комплекса представлен на рис. 1. При разметке изображений генерировалась следующая информация: расположение граничащего прямоугольника (BoundingBox) и класс, к которому принадлежит объект. Каждой марке автомобиля было присвоено свое числовое значение.

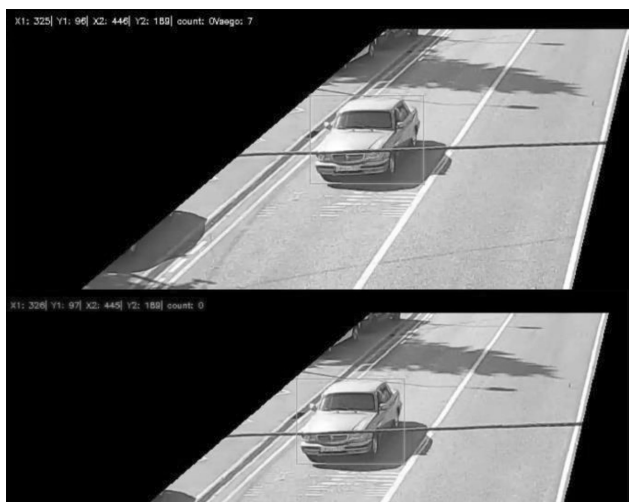


Рис. 1. Результат разметки изображения

Fig 1. Image markup result

Каждый раз при разметке нового изображения информация записывалась в новый файл с ответами. Для упрощения последующей обработки файлу присваивалось имя с текущей датой, также после разметки изображение меняло свое название на такое же имя, что и файл ответов.

После этого происходит предварительная обработка изображения, а именно, изменяется размер изображения, поскольку нейронная сеть принимает изображения фиксированного входного размера. Перед подачей в нейронную сеть из изображения формируется массив. После чего данные нормализуются. Так как изображение – это матрица чисел в диапазоне от 0

до 255, то данные преобразуются к в диапазону от 0,0 до 1,0, таким образом обучение происходит быстрее и результат оказывается более точным.

Описание модели разработанной нейронной сети

За основу архитектуры нейронной сети, которая использовалась для решения поставленной задачи, была взята архитектура tiny-YOLO-InceptionResNet [Zoev et al., 2019, Khokhlov et al., 2020]. Tiny-YOLO-InceptionResNet – это модификация модели нейронной сети tiny-YOLO.

Чтобы оптимизировать скорость работы, было принято решение обучить 2 модели нейронной сети, после чего запускать их в разных потоках.

Архитектура разработанной нейронной сети представлена в табл. 1.

Общее количество обучающихся и необучающихся параметров:

- Total params – 1,702,421;
- Trainable params – 1,702,421;
- Non-trainable params – 0.

ОПИСАНИЕ СЛОЕВ

Conv2D

Это слой 2D-свертки, который создает ядра свертки. После операции свертки на выходе формируется выходная карта (матрица) признаков – output feature maps.

В основном, первый сверточный слой изучает небольшие локальные шаблоны, такие как края. Второй – более крупные шаблоны, состоящие из признаков, возвращаемых первым слоем и т.д. По мере обучения нейросетевой модели производился вывод тепловых карт активации класса на изображении. Таким образом можно понять, как слои сверточной нейронной сети преобразуют входные данные, а также понять значение каждого фильтра сверточной нейронной сети. Помимо этого визуализация фильтров помогает точно понять шаблоны или визуальные концепции, которые может легко принять каждый фильтр в сверточной нейронной сети.

MaxPooling2D

Операция подвыборки (pooling) уменьшает размер карты признаков, используя некоторую функцию для объединения ближайших элементов матрицы. В качестве функции может использоваться среднее арифметическое или максимум.

UpSampling2D

Эта функция увеличивает размерность изображения путем повторения строк и столбцов.

Flatten

Слой Flatten используется для «сглаживания» входных данных, то есть для ввода, который обычно используется при переходе от сверточного слоя к полносвязному слою. Сглаживание не влияет на размер подаваемых данных. Входящий двумерный вектор преобразуется в одномерный.

Архитектура нейронной сети Tiny YOLO XY [Neural network architecture Tiny YOLO XY]

№ компонента архитектуры нейронной сети [Neural network architecture component number]	Описание [Description]
1	Три сверточных слоя, функция активации – RELU, параметр заполнения – same [Three convolution layers, activation function – RELU, filling parameter – same]
2	Слой подвыборки (функция max, окно подвыборки 2×2) [Pooling layer (max function, pooling window 2×2)]
3	Три сверточных слоя, функция активации – RELU, параметр заполнения – same [Three convolution layers, activation function – RELU, filling parameter – same]
4	Слой подвыборки (функция max, окно подвыборки 2×2) [Pooling layer (max function, pooling window 2×2)]
5	Два сверточных слоя, функция активации – RELU, параметр заполнения – same [Two convolution layers, activation function – RELU, filling parameter – same]
6	Слой подвыборки (функция max, окно подвыборки 2×2) [Pooling layer (max function, pooling window 2×2)]
7	Два сверточных слоя, функция активации – RELU, параметр заполнения – same [Two convolution layers, activation function – RELU, filling parameter – same]
8	Слой подвыборки (функция max, окно подвыборки 2×2) [Pooling layer (max function, pooling window 2×2)]
9	Два сверточных слоя, функция активации – RELU, параметр заполнения – same [Two convolution layers, activation function – RELU, filling parameter – same]
10	Слой подвыборки (функция max, окно подвыборки 2×2) [Pooling layer (max function, pooling window 2×2)]
11	Два сверточных слоя, функция активации – RELU, параметр заполнения – same [Two convolution layers, activation function – RELU, filling parameter – same]
12	Слой подвыборки (функция max, окно подвыборки 2×2) [Pooling layer (max function, pooling window 2×2)]
13	Два сверточных слоя, функция активации – RELU, параметр заполнения – same [Two convolution layers, activation function – RELU, filling parameter – same]
14	Слой увеличения размера массива (окно 2×2) [Layer increasing array size (window 2×2)]
15	Слой преобразования данных из 2D-представления в плоское [Data conversion layer from 2D to data vector]
16	Слой Dropout для борьбы с переобучением нейронной сети [Dropout layer to combat retraining of neural network]
17	Полносвязный слой с четырьмя нейронами [Fully connected layer with four neurons]

Dropout

Это один из методов регуляризации (борьбы с переобучением модели). Он используется, если происходит переобучение. На каждом этапе обучения весов нейронных связей некоторая случайным образом выбранная часть весов «замораживается» и не подвергается изменениям. Процент таких связей определяется заранее. Это позволяет избежать «привыкания» нейронной сети к конкретному набору данных, и делает сеть более гибкой в обучении. Применение процедуры дропаута позволяет работать с задачами, где обучающая выборка отличается по количественному составу от реальных

данных, где существуют вариации, немного отличающиеся от данных, встреченных в обучающем наборе. Точность определения таких «немного измененных» данных при этом повышается.

Если процедура дропаута не применяется, возможна ситуация, когда нейронная сеть «запоминает», «заучивает наизусть» только тренировочную выборку, но не может выделить характеристики всего многообразия реальных данных. Слой Dropout позволяет предотвратить эту ситуацию. В таком случае обучается модель медленнее и на обучающей выборке показывает результаты хуже, чем без применения слоя Dropout, зато на тестовой выборке точность повышается.

Dense

Полносвязный слой позволяет построить нелинейную функцию, ставящую в соответствие конкретному классу набора карт активации для классификации объекта. Этот слой также имеет нейроны связей, что позволяет настраивать функцию определения принадлежности к классу.

Функция активации **ReLU** используется для всех скрытых слоев, за исключением последнего полносвязного слоя. Функция требует меньше вычислительных ресурсов, что ускоряет физическое время обучения нейронной сети.

Функция активации ReLU: $A(x) = \max(0, x)$. ReLU возвращает значение x , если x положительно, и 0 если отрицательно или равно нулю.

Также, используя ReLU, мы можем инактивировать некоторые нейроны, это сделает нейронную сеть более быстрой в пересчете времени на одну эпоху обучения.

У функции есть существенный недостаток, который приводит к тому, что обучение в некоторых частях нейронной сети останавливается необратимо. ReLU – кусочно заданная функция, которая при отрицательных значениях аргумента равна 0, что означает невозможность применения к ней в таком случае метода градиентного спуска (метода, приведшего к колоссальному распространению нейронных сетей), так как градиент нуля всегда равен нулю и веса не корректируются. Даже при очень большой ошибке обучение не будет производиться и веса останутся прежними. Такое явление называется **проблемой умирающего ReLU** (Dying ReLU problem) или проблемой затухающего градиента. Та часть нейронной сети, где произошла подобная ситуация, будет оставаться не обученной.

Однако можно избежать данной проблемы, если применить вариацию функции RELU, хотя это потребует одного дополнительного арифметического действия. Положим, что в части, где аргумент отрицательный, функция возвращает значение $x \cdot 0,01$, а не 0. В таком случае градиент перестает быть равным нулю, и это позволяет избежать ситуации «атрофии» некоторой части нейронной сети.

Для измерения метрики использовались следующие параметры:

- функция потерь: `loss='mean_squared_error'`;
- оптимизатор: `optimizer=optimizers.RMSprop (learning_rate=0.000073147)`;
- метрика: `metrics=['accuracy']`.

Функция потерь – среднеквадратическая ошибка (MSE)

Среднеквадратическая ошибка рассчитывается как среднее квадратов разностей между прогнозируемыми и фактическими значениями. Результат всегда положительный, независимо от знака, прогнозируемого и фактического значения, а идеальное значение равно 0,0. Возведение в квадрат означает, что большие ошибки приводят к большему размеру параметра, чем меньшие ошибки, а это означает, что модель наказывается за большие ошибки.

RMSprop

Оптимизатор – среднеквадратичное распространение корня (RMSprop) – это экспоненциально затухающее среднее значение [Dongro Xu et al., 2021]. Оптимизатор более чувствителен к ограничению градиента последних шагов, чем к сумме прошлых градиентов.

В документации к фреймворку Keras рекомендуется не изменять параметры оптимизатора. Но т.к. при обучении сети со стандартными параметрами оптимизатор показал плохие результаты, было принято решение изменить скорость обучения. Методом подбора нужной скорости, был выявлен коэффициент скорости обучения `learning_rate = 0,000073147`. При таком коэффициенте во время обучения нейронная сеть не попадала в локальный минимум или максимум.

Метрика Accuracy

Поскольку в реализованной архитектуре нужно было предсказывать числа, была использована метрика Accuracy которая хорошо подходит для задачи регрессии.

Accuracy – этот показатель, позволяет оценить точность работы модели классификации по всем классам объектов. Такая метрика чаще всего применяется, если каждый класс имеет одинаковую важность по сравнению с другими. Для расчета параметра необходимо вычислить отношение правильных прогнозов к общему количеству прогнозов. Метрика может давать недостоверные результаты, если набор данных несбалансирован по всем классам.

ЭКСПЕРИМЕНТ И ОЦЕНКА ОБУЧЕНИЯ

При обучении архитектуры для поиска границ объектов были использованы следующие гиперпараметры с о следующими значениями:

- размер пакета для обучения `batch_size = 75`;
- количество эпох обучения `epochs = 800`;
- объем тестовой выборки (в долях) `validation_split = 0.2`.

Обучение проходило в 2 этапа.

Первый этап обучался на 2032 изображениях на протяжении 500 эпох. На тот момент с существующими данными точность работы на проверочных данных составляла: 97,64% и распознавание объекта на одном кадре длилось около 200 мс; После обучения во время тестирования сеть показала результат на тестовых данных равный 44%.

Вторым этапом в обучении сети для обнаружения границ объектов было дообучение. В набор данных добавилось еще около 1026 изображений и проходило на уже на 300 эпохах. Но при этом точность предсказания уменьшилась до 96,44%. Это явление было вызвано тем, что набор данных был неоднородных: изображения были размечены с несколькими вариациями: оригинальное изображение, изображение в котором отсеяна лишняя информация (на дороге видна только одна полоса движения) и изображение, где видно две полосы движения. Пример такой неоднородности представлен на рис. 2.



Рис. 2. Неоднородность данных

Fig. 2. Data heterogeneity

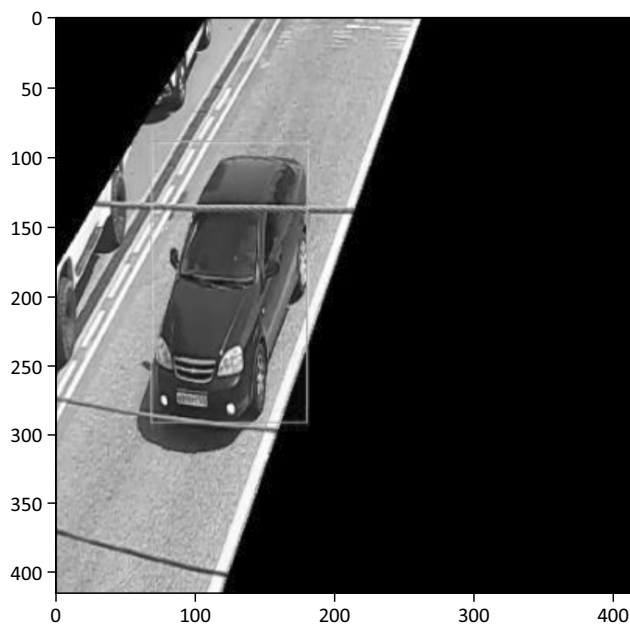


Рис. 3. Результат работы на тестовых изображениях

Fig. 3. The result of work on test images

После дообучения во время тестирования сеть показала результат на тестовых данных, точность равна 88%.

Результаты работы нейронной сети для обнаружения границ транспортного средства (bounding box) можно увидеть на рис. 3.

Как можно увидеть нейронная сеть довольно точно определяет границы объекта.

При обучении нейронной сети использовались фреймворки и библиотеки: TensorFlow 2, Keras, OpenCV, Matplotlib, NumPy, разработка производилась на языке программирования Python.

Литература

1. Atibi M., Atouf I., Boussaa M., Bennis A. Real-time detection of vehicles using the haar-like features and artificial neuron networks // *Procedia Computer Science*. 2015. Vol. 73. Pp. 24–31. ISSN 1877-0509. URL: <https://doi.org/10.1016/j.procs.2015.12.044>.
2. Dongpo Xu, Shengdong Zhang, Huisheng Zhang, Danilo P. Mandic. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization // *Neural Networks*. 2021. Vol. 139. Pp. 17–23. ISSN 0893-6080. URL: <https://doi.org/10.1016/j.neunet.2021.02.011>.

ВЫВОДЫ

В данной работе описывается модель, полученная с применением нейросетевого подхода для определения границ автотранспортного средства из видео потока, которая может быть использована для предварительной работы по распознаванию и отслеживанию автомобилей при помощи камеры видеонаблюдения. Созданная сверточная нейронная сеть обладает точностью 88%. Выбор архитектуры нейронной сети (вариации архитектуры Tiny-YOLO-InceptionResNet) оправдал себя и показал достаточную точность, которую авторы планируют в дальнейших экспериментах повысить еще.

Описанная нейро-сетевая модель может быть частью программного комплекса для решения более локальных задач идентификации транспортного средства.

ЗАКЛЮЧЕНИЕ

В работе представлен метод создания нейро-сетевой модели определения границ автотранспортного средства из видео потока. Для обучения модели использовался набор данных сформированный из видео потока с камеры, направленной на полосу дорожного движения. Набор данных был предварительно обработан, в частности обрезаны лишние детали, информация нормализована, данные были размечены при помощи специально созданного программного комплекса.

В ходе работы авторами была предложена нейро-сетевая модель, обучена и протестирована. Точность полученной модели сверточной нейронной сети достигает 88%. В дальнейшем модель может стать частью программного комплекса для роботизированной автомойки.

References

1. Atibi M., Atouf I., Boussaa M., Bennis A. Real-time detection of vehicles using the haar-like features and artificial neuron networks. *Procedia Computer Science*. 2015. Vol. 73. Pp. 24–31. ISSN 1877-0509. URL: <https://doi.org/10.1016/j.procs.2015.12.044>.
2. Dongpo Xu, Shengdong Zhang, Huisheng Zhang, Danilo P. Mandic. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. *Neural Networks*. 2021. Vol. 139. Pp. 17–23. ISSN 0893-6080. URL: <https://doi.org/10.1016/j.neunet.2021.02.011>.

3. *Fan Q., Brown L., Smith J.* A closer look at Faster R-CNN for vehicle detection // *IEEE Intelligent Vehicles Symposium*. 2016. Vol. IV. Pp. 124–129. DOI: 10.1109/IVS.2016.7535375.
4. *Hoanh Nguyen.* Improving faster R-CNN framework for fast vehicle detection // *Mathematical Problems in Engineering*. 2019. Article ID: 3808064. 11 p. URL: <https://doi.org/10.1155/2019/3808064>
5. *Khokhlov et al.* Tiny-YOLO object detection supplemented with geometrical data // *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 2020. Pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9128749.
6. *Laroca R. et al.* A Robust real-time automatic license plate recognition based on the YOLO detector // *International Joint Conference on Neural Networks (IJCNN)*. 2018. Pp. 1–10. DOI: 10.1109/IJCNN.2018.8489629.
7. *Maity M., Banerjee S., Sinha Chaudhuri S.* Faster R-CNN and YOLO based Vehicle detection: A survey // 5th International Conference on Computing Methodologies and Communication (ICCMC). 2021. Pp. 1442–1447. DOI: 10.1109/ICCMC51019.2021.9418274.
8. *Manana M., Tu C., Owolawi P.A.* Preprocessed faster RCNN for vehicle detection // *International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. 2018. Pp. 1–4. DOI: 10.1109/ICONIC.2018.8601243.
9. *Miao Y., Liu F., Hou T. et al.* A nighttime vehicle detection method based on YOLO v3 // *Chinese Automation Congress (CAC)*. 2020. Pp. 6617–6621. DOI: 10.1109/CAC51589.2020.9326819.
10. *Rybski P.E., Huber D., Morris D.D., Hoffman R.* Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features // *IEEE Intelligent Vehicles Symposium*. 2010. Pp. 921–928. DOI: 10.1109/IVS.2010.5547996.
11. *Shi K., Bao H., Ma N.* Forward vehicle detection based on incremental learning and fast R-CNN // 13th International Conference on Computational Intelligence and Security (CIS). 2017. Pp. 73–76. DOI: 10.1109/CIS.2017.00024.
12. *Shin H.-C. et al.*, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning // *IEEE Transactions on Medical Imaging*. 2016. Vol. 35. No. 5. Pp. 1285–1298. DOI: 10.1109/TMI.2016.2528162.
13. *Xu Y., Yu G., Wu X. et al.* An enhanced Viola–Jones vehicle detection method from unmanned aerial vehicles imagery // *IEEE Transactions on Intelligent Transportation Systems*. 2017. Vol. 18. No. 7. Pp. 1845–1856. DOI: 10.1109/TITS.2016.2617202.
14. *Zehang Sun, Bebis G., Miller R.* On-road vehicle detection using Gabor filters and support vector machines // 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No. 02TH8628). 2002. Vol. 2. Pp. 1019–1022. DOI: 10.1109/ICDSP.2002.1028263.
15. *Zoev V., Beresnev A.P., Markov N.G.* Convolutional neural networks of the YOLO class in computer vision systems for mobile robotic complexes // *International Siberian Conference on Control and Communications (SIBCON)*. 2019. Pp. 1–5. DOI: 10.1109/SIBCON.2019.8729605.
16. *Катермина Т.С., Сибгатулин А.Ф.* Применение методов искусственного интеллекта к задаче диагностики заболеваний дыхательных путей // *Computational Nanotechnology*. 2022. Т. 9. № 2. С. 92–103. DOI: 10.33693/2313-223X-2022-9-2-92-103.
17. *Разинкин В.Б., Катермина Т.С.* Распознавание лица по фотографии // *International Journal of Advanced Studies*. 2018. Т. 8. № 1-2. С. 171–180.
3. *Fan Q., Brown L., Smith J.* A closer look at Faster R-CNN for vehicle detection. *IEEE Intelligent Vehicles Symposium*. 2016. Vol. IV. Pp. 124–129. DOI: 10.1109/IVS.2016.7535375.
4. *Hoanh Nguyen.* Improving faster R-CNN framework for fast vehicle detection. *Mathematical Problems in Engineering*. 2019. Article ID: 3808064. 11 p. URL: <https://doi.org/10.1155/2019/3808064>
5. *Khokhlov et al.* Tiny-YOLO object detection supplemented with geometrical data. *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 2020. Pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9128749.
6. *Laroca R. et al.* A Robust real-time automatic license plate recognition based on the YOLO detector. *International Joint Conference on Neural Networks (IJCNN)*. 2018. Pp. 1–10. DOI: 10.1109/IJCNN.2018.8489629.
7. *Maity M., Banerjee S., Sinha Chaudhuri S.* Faster R-CNN and YOLO based Vehicle detection: A survey. 5th International Conference on Computing Methodologies and Communication (ICCMC). 2021. Pp. 1442–1447. DOI: 10.1109/ICCMC51019.2021.9418274.
8. *Manana M., Tu C., Owolawi P.A.* Preprocessed faster RCNN for vehicle detection. *International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. 2018. Pp. 1–4. DOI: 10.1109/ICONIC.2018.8601243.
9. *Miao Y., Liu F., Hou T. et al.* A nighttime vehicle detection method based on YOLO v3. *Chinese Automation Congress (CAC)*. 2020. Pp. 6617–6621. DOI: 10.1109/CAC51589.2020.9326819.
10. *Rybski P.E., Huber D., Morris D.D., Hoffman R.* Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features. *IEEE Intelligent Vehicles Symposium*. 2010. Pp. 921–928. DOI: 10.1109/IVS.2010.5547996.
11. *Shi K., Bao H., Ma N.* Forward vehicle detection based on incremental learning and fast R-CNN. 13th International Conference on Computational Intelligence and Security (CIS). 2017. Pp. 73–76. DOI: 10.1109/CIS.2017.00024.
12. *Shin H.-C. et al.*, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*. 2016. Vol. 35. No. 5. Pp. 1285–1298. DOI: 10.1109/TMI.2016.2528162.
13. *Xu Y., Yu G., Wu X. et al.* An enhanced Viola–Jones vehicle detection method from unmanned aerial vehicles imagery. *IEEE Transactions on Intelligent Transportation Systems*. 2017. Vol. 18. No. 7. Pp. 1845–1856. DOI: 10.1109/TITS.2016.2617202.
14. *Zehang Sun, Bebis G., Miller R.* On-road vehicle detection using Gabor filters and support vector machines. 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No. 02TH8628). 2002. Vol. 2. Pp. 1019–1022. DOI: 10.1109/ICDSP.2002.1028263.
15. *Zoev V., Beresnev A.P., Markov N.G.* Convolutional neural networks of the YOLO class in computer vision systems for mobile robotic complexes. *International Siberian Conference on Control and Communications (SIBCON)*. 2019. Pp. 1–5. DOI: 10.1109/SIBCON.2019.8729605.
16. *Katermina T.S., Sibagatulin A.F.* Application of artificial intelligence methods to the task of diagnosing respiratory diseases. *Computational Nanotechnology*. 2022. Vol. 9. No. 2. Pp. 92–103. DOI: 10.33693/2313-223X-2022-9-2-92-103. (In Rus.)]
17. *Razinkin V.B., Katermina T.S.* Face recognition by photography. *International Journal of Advanced Studies*. 2018. Vol. 8. No. 1-2. Pp. 171–180. (In Rus.)]

Статья проверена программой Антиплагиат. Оригинальность – 85,55%

Р е ц е н з е н т: *Казиахмедов Т.Б.*, кандидат педагогических наук, доцент; доцент, заведующий кафедрой информатики и методики преподавания информатики Нижневартковского государственного университета

Статья поступила в редакцию 14.08.2022, принята к публикации 19.09.2022

The article was received on 14.08.2022, accepted for publication 19.09.2022

СВЕДЕНИЯ ОБ АВТОРАХ

Катермина Татьяна Сергеевна, кандидат технических наук; доцент кафедры информатики и методики преподавания информатики Нижневартковского государственного университета. Нижневартовск, Ханты-Мансийский автономный округ – Югра, Российская Федерация. Author ID: 659022; ScopusAuthor ID: 57160399700; Researcher ID: N-7840-2016; ORCID: 0000-0003-4637-8861; E-mail: nggu-lib@mail.ru

Лазоренко Евгений Викторович, магистрант Нижневартковского государственного университета. Нижневартовск, Ханты-Мансийский автономный округ – Югра, Российская Федерация. E-mail: rolaraltis@hotmail.com

ABOUT THE AUTHORS

Tatyana S. Katermina, Cand. Sci. (Eng.); associate professor at the Department of Informatics and Methods of Teaching Informatics of the Nizhnevartovsk State University. Nizhnevartovsk, Khanty-Mansi Autonomous Okrug – Yugra, Russian Federation. Author ID: 659022; ScopusAuthor ID: 57160399700; Researcher ID: N-7840-2016; ORCID: 0000-0003-4637-8861; E-mail: nggu-lib@mail.ru

Evgenij V. Lazorenko, student at the Nizhnevartovsk State University. Nizhnevartovsk, Khanty-Mansi Autonomous Okrug – Yugra, Russian Federation. E-mail: rolaraltis@hotmail.com