

ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ
INFORMATION TECHNOLOGIES AND TELECOMMUNICATION

УДК 004.932

DOI:10.18413/2518-1092-2017-2-4-50-58

Чуйков Р.Ю.
Юдин Д.А.

ОБНАРУЖЕНИЕ ТРАНСПОРТНЫХ СРЕДСТВ НА ИЗОБРАЖЕНИЯХ
ЗАГОРОДНЫХ ШОССЕ НА ОСНОВЕ МЕТОДА
SINGLE SHOT MULTIBOX DETECTOR

Белгородский государственный технологический университет им. В.Г. Шухова, ул. Костюкова, д. 46,
г. Белгород, 308012, Россия

e-mail: chuykov95@mail.ru, yuddim@yandex.ru

Аннотация

В статье рассмотрено применение современного метода обнаружения объектов на изображении – Single Shot Multibox Detector. Осуществлено обучение свёрточной нейронной сети для обнаружения транспортных средств на выборке из 3000 изображений с размеченными областями расположения автомобилей. Произведена проверка качества работы сети на 7000 тестовых изображениях. Тестовая и обучающая выборки содержат изображения, сделанные монокулярной камерой, установленной в транспортном средстве, движущемся по загородным шоссе в светлое время суток. Полнота и точность обнаружения объектов на тестовой выборке составляет соответственно более 88% и 78%. Распознавание одного кадра занимает 28,5 миллисекунд, при этом вычисления осуществляются на графическом процессоре с использованием технологии NVidia CUDA. Полученные результаты могут быть применены в системах помощи водителю и мониторинга дорожной обстановки.

Ключевые слова: распознавание изображений; глубокое обучение; свёрточная нейронная сеть; обнаружение; транспортное средство.

UDC 004.932

Chuykov R.Y.
Yudin D.A.

VEHICLE DETECTION ON HIGHWAY IMAGES BASED
ON SINGLE SHOT MULTIBOX DETECTOR

Belgorod State Technological University n.a. V.G. Shukhov, 46 Kostukova St., Belgorod, 308012, Russia

e-mail: chuykov95@mail.ru, yuddim@yandex.ru

Abstract

In this article we consider the application of the modern object detection method – Single Shot Multibox Detector. We have trained the convolutional neural network for vehicle detection on a sample of 3000 images with marked areas where are the vehicles are placed. A network quality check was performed on 7000 test images. The test and training samples contain images made by a monocular camera mounted in a vehicle moving along suburban highways during daylight hours. Recall and precision of object detection on the test sample is correspondingly more than 88% and 78%. Recognition of one frame takes 28.5 milliseconds. Experiment was performed on a graphics processor using NVidia CUDA technology. The obtained results can be applied in driver assistance systems and monitoring of the traffic situations.

Keywords: image recognition; deep learning; convolutional neural network; detection; vehicle.

ВВЕДЕНИЕ

Обнаружение транспортных средств на изображениях является необходимым элементом систем помощи водителю, мониторинга дорожной обстановки, беспилотного управления автомобилями. Решение этой задачи может сделать возможным информирование водителя о текущей обстановке на дороге, например, о приближающихся автомобилях или других участниках дорожного движения. Транспортные средства имеют большое разнообразие типов, отличающихся цветом и формой, поэтому на практике задача их обнаружения и классификации на дороге на основе анализа изображений до сих пор остается открытой.

За последние годы достигнут значительный прогресс в области обнаружения объектов с использованием сверточных нейронных сетей. Современные детекторы на основе этих сетей, такие как R-FCN [9], Faster R-CNN [3], Multibox [12], SSD [15] и YOLO [20] стали достаточно быстрыми для использования в потребительских продуктах и для работы на мобильных и встраиваемых устройствах [14].

В методе Faster R-CNN [5] обнаружение объектов проходит в два этапа. На первом этапе с помощью сети прогнозирования регионов (*Region proposal network, RPN*), представляющей собой сверточную нейронную сеть (например, архитектуры VGG-16 [13], ResNet [4] и др.) автоматически извлекаются признаки изображения и делаются предположения о возможных местах расположения объекта. На втором этапе, называемом классификатором областей (*Box classifier*), каждый из найденных регионов вырезается и классифицируется с помощью еще одной сверточной нейронной сети, при этом также происходит уточнение формы прямоугольников.

Метод R-FCN [10] был предложен для ускорения Fast R-CNN, который требует применения громоздкого выходного классификатора несколько сотен раз. В методе R-FCN (*Region-based Fully Convolutional Networks*), в отличие от Fast R-CNN обрезка областей не происходит на выходе сети прогнозирования регионов, вместо этого к выходу первой сети добавляются сверточные слои для дополнительного извлечения признаков и обрезка областей производится из последнего сверточного слоя. Далее происходит классификация с помощью всего лишь одного или двух полносвязных слоев нейронов. Такой подход позволил достичь точности сравнимой с Faster R-CNN при более быстром времени работы. Недавно модель R-FCN была также приспособлена для задачи сегментации в модели TA-FCN [18], которая выиграла задачу сегментации экземпляров COCO [7] 2016 года.

Метод *Single Shot Detector (SSD)* [15] был опубликован сравнительно недавно. Термин SSD используется для описания архитектур, в которых используется одна сверточная нейронная сеть (*feedforward convolutional network*) для непосредственного предсказания расположения областей и их классов, без применения второго этапа классификации. В этом методе на выходе нейронной сети формируются несколько тысяч прогнозов для возможных регионов расположения объектов разной формы на разных масштабах, затем с помощью подавления немаксимумов (*Non-Maximum Suppression*) происходит выбор нескольких наиболее вероятных областей. Такая единая структура, одновременно с учетом различных масштабов изображения обеспечила методу SSD наиболее высокие показатели по скорости и качеству обнаружения объектов по сравнению с остальными современными подходами [14].

Так же для распознавания объектов на изображениях может применяться быстродействующий метод Вилы-Джонса [1, 19, 21]. Метод использует скользящее окно, которое движется с некоторым шагом по изображению, и с помощью каскадов Хаара, определяет, есть ли в данной области объект. Этот метод имеет такие преимущества как детектирование нескольких объектов на изображении и хорошая скорость обнаружения. Однако этот метод имеет длительное время обучения, а также характеризуется ограниченными возможностями описания возможностей объектов с помощью признаков Хаара. Поэтому метод Виолы-Джонса не позволяет построить качественный детектор сложных объектов (к которым относятся автомобили под разными ракурсами), имеющий возможность дополнительного обучения.

ПОСТАНОВКА ЗАДАЧИ

В настоящей работе рассматривается обнаружение транспортных средств на изображениях загородных шоссе на основе метода Single Shot Multibox Detector (SSD). Обучающая и тестовая выборки соответственно содержат 3000 и 7000 изображений, сделанные монокулярной камерой, установленной в транспортном средстве, движущемся по загородным шоссе в светлое время суток. На каждом изображении размечены области расположения автомобилей. Изображения взяты из набора данных открытого конкурса по распознаванию автомобилей, проводившегося российской компанией «Когнитивные технологии» [2] в 2015 году.

На рисунке 1 изображен пример разметки элемента обучающей выборки, на нем показано изображение и соответствующий ему файл разметки, который содержит такие данные как: имя файла изображения, его размер, имя класса, координаты верхнего левого и нижнего правого прямоугольника, содержащего объект.

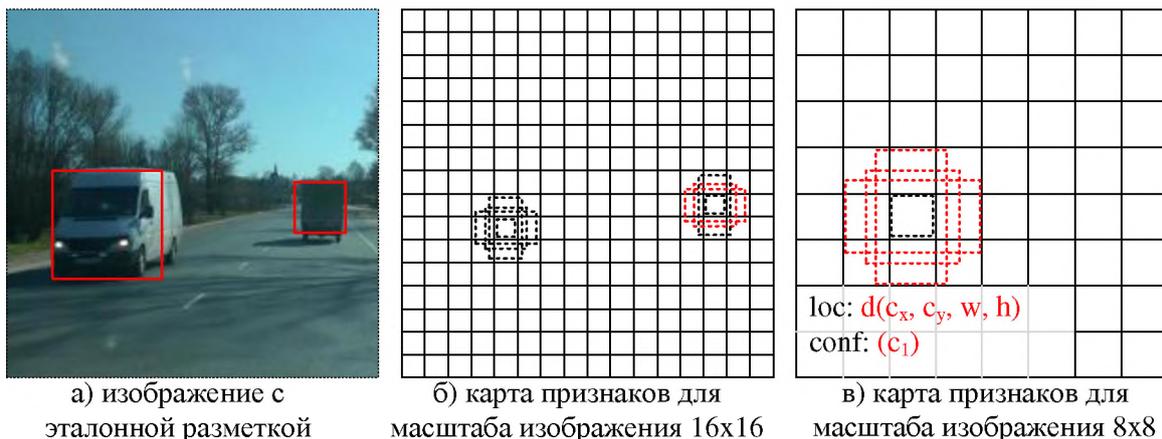


Рис. 1. Пример эталонной разметки изображений обучающей и тестовой выборки
Fig. 1. An example of the reference marking of the images of the training and test samples

ПРИМЕНЕНИЕ МЕТОДА SSD ДЛЯ ОБНАРУЖЕНИЯ ТРАНСПОРТНЫХ СРЕДСТВ

В настоящей статье исследуется применение метода SSD (Single Shot MultiBox Detector) на примере модели SSD 300 [15]. На вход этой модели подается изображение 300x300 пикселей, затем к изображению применяется сверточные слои из усеченной стандартной модели VGG-16 (не используются выходные классифицирующие слои), далее к выходному слою добавляются специальные сверточные слои, представляющие собой изображение в разных масштабах. Пространственная размерность убывает до тех пор, пока не станет равной единице. Каждый из специальных сверточных слоев позволяет сформировать карту признаков для разных масштабов изображения (см. рисунок 2), в которой для пикселя карты определяется какой из ограничивающих прямоугольников в области 3×3 лучше всего совпадает с эталонной разметкой. Все эти карты объединяются в единый выходной слой, содержащий информацию о 8732 регионов (прямоугольников), в которых может находиться объект [15].

Для рассматриваемой задачи каждый из этих регионов содержит информацию о прогнозе класса объекта c_1 (рассматривается только один класс – «транспортное средство»), о корректировке шаблонных прямоугольников по координатам c_x и c_y , а также по ширине и высоте w и h (рисунок 2). Итоговые области выбираются из этих 8732 прямоугольников с помощью метода подавления немаксимумов.



а) изображение с эталонной разметкой

б) карта признаков для масштаба изображения 16x16

в) карта признаков для масштаба изображения 8x8

Рис. 2. Пояснения к обучению детектора объектов SSD: а – для обучения SSD требует только изображение и эталонная разметка в виде прямоугольников, б, в – для каждой размерности карты признаков (например, 16x16 и 8x8 в (б) и (в)) строятся шаблонные прямоугольники (например, по 4 для каждого из пикселя карты), и для каждого из них указывается принадлежность к классу c_1 , а также величины корректировки шаблонных прямоугольников по координатам c_x и c_y , а также по ширине и высоте w и h , чтобы они совпадали с эталонными

Fig. 2. Explanations for learning the detector of objects SSD: а – SSD only needs an input image and ground truth boxes for each object during training, б, в – in a convolutional fashion, we evaluate a small set of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 16x16 and 8x8 в (б) and (в)). For each default box, we predict both the shape offsets and the confidences for all object categories c_1 . At training time, we first match these default boxes to the ground truth boxes

Реализация метода SSD [15] осуществлена на языке программирования Python 3 с использованием библиотек глубокого обучения Tensor Flow [16] и Keras [6], а также технологии NVIDIA CUDA [3].

На вход обучающего алгоритма подается обучающая выборка размером 3000 изображений, pickle-файл [8] с разметкой этих изображений содержащий информацию об имени файла изображения, классах объектов и их положении. Затем алгоритм проходит 30 итераций (эпох), сохраняя при этом файл с весами для каждой итерации. При необходимости метод позволяет загрузить веса уже обученной модели для тренировки сети на новых изображениях.

При тестировании на вход алгоритма SSD подаются веса обученной модели, набор изображений и файлы разметки в формате xml для вычисления метрик. Алгоритм находит все объекты на изображении, для каждого объекта создается массив с именем класса, координатами прямоугольника и вероятностью обнаружения.

Обучение и тестирование модели SSD было произведено на компьютере со следующими характеристиками: процессор Intel Core i7-4790 3.60GHz; оперативная память 16,0 ГБ; видеокарта MSI Nvidia GeForce GTX 1060, частота графического процессора 1594 МГц, 6144 Мб видеопамати GDDR5, частота видеопамати 8100 МГц, разрядность шины видеопамати 192 бит, число универсальных процессоров 1280.

РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

В ходе работы проведены вычислительные эксперименты для трех различным образом обученных свёрточных нейронных сетей:

1) свёрточная нейронная сеть модели SSD 300, уже обученная на выборке Pascal VOC [11] (назовём ее SSD300_VOC),

2) сверточная нейронная сеть из [11], дополнительно обученная на имеющейся обучающей выборке из 3000 изображений (назовём ее SSD300_VOC_Tuned),

3) сверточная нейронная сеть модели SSD 300, обученная с нуля на имеющейся обучающей выборке из 3000 изображений (назовём ее SSD300_Cars).

Результаты вычислительных экспериментов по обучению и тестированию нейронных сетей приведены в таблице.

Таблица

Результаты обнаружения транспортных средств на обучающей и тестовой выборке на основе модели SSD 300

Table

The results of vehicle detection on the training and test sample based on the SSD 300 model

Свёрточная нейронная сеть	Обучающая выборка из 3000 изображений		Тестовая выборка из 7000 изображений	
	Полнота (R)	Точность (P)	Полнота (R)	Точность (P)
SSD300_VOC	0.475	0.889	0.727	0.597
SSD300_VOC_Tuned	0.984	0.990	0.888	0.785
SSD300_Cars	0.317	0.857	0.665	0.564

Для оценки качества обнаружения транспортных средств используются такие метрики как мера пересечения найденных и эталонных прямоугольников, содержащих транспортное средство (Intersection, I), полнота (Recall, R) и точность (Precision, P) обнаружения объектов [9]. Рассмотрим каждый показатель отдельно.

Мера пересечения найденных и эталонных прямоугольников I (1) показывает, как точно свёрточная нейронная сеть нашла прямоугольник относительно прямоугольника эталонной разметки (рисунок 3).

$$I = \frac{S_I}{S_f + S_{gt} - S_I}, \quad (1)$$

где S_I – площадь пересечения истинного и вычисленного прямоугольника, S_f – площадь найденного алгоритмом прямоугольника, S_{gt} – площадь эталонного прямоугольника (ground truth).

Полнота R (2) показывает чувствительность алгоритма к ошибкам 2-го рода, то есть, пропускам, и равна отношению количества правильно найденных объектов к общему количеству этих объектов в эталонной разметке.

$$R = \frac{tp}{tp + fn}, \quad (2)$$

где tp – истинно-положительные (true positives) – те объекты, которые мы ожидали увидеть и получили на выходе, fn – ложно-отрицательные (false negatives) – объекты, которые мы ожидали увидеть, но алгоритм их не определил (пропуски).

Точность P (3) показывает чувствительность алгоритма к ошибкам 1-го рода, то есть, ложным срабатываниям и равна отношению количества правильно найденных объектов к общему количеству найденных алгоритмом прямоугольников.

$$P = \frac{tp}{tp + fp}, \quad (3)$$

где fp – ложно-положительные (false positives) объекты – такие, которых быть на выходе не должно, но алгоритм их ошибочно вернул на выходе (ложные срабатывания).

Все метрики выбраны с учетом общепринятых подходов, изложенных в работе [17].

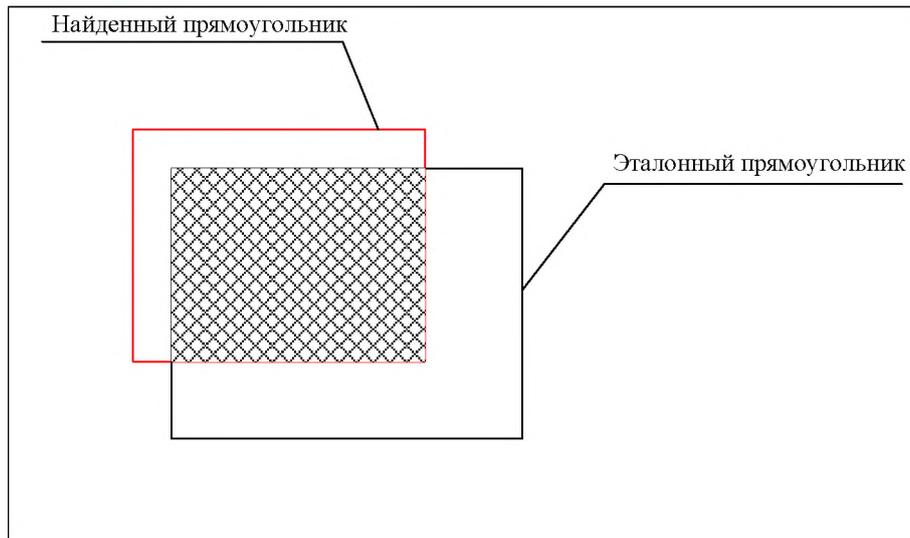


Рис. 3. Пояснение к определению меры пересечения I найденных и эталонных прямоугольников
Fig. 3. Explanation to the definition of the measure of intersection I of the found and reference rectangles

На основе анализа результатов, приведенных в таблице, можно сказать, что наилучшие показатели качества у сети SSD300_VOC_Tuned, обученной на изображениях с автомобилями из имеющейся выборки, при этом веса настраивались не с нуля, а с использованием весов сети из [11] заранее обученной на изображениях из набора Pascal VOC. На втором месте готовая сеть SSD300_VOC из [11], не обучавшаяся на обучающей выборке, а на третьем обученная сеть с нуля SSD300_Cars.

Такие результаты отражают факт, что чем больше различных изображений автомобилей (объектов) было использовано при обучении нейронной сети, тем выше результаты и качества распознавания. Также результат доказывает перспективность и необходимость пополнения обучающей выборки для создания качественных алгоритмов обнаружения объектов в различных условиях съемки.

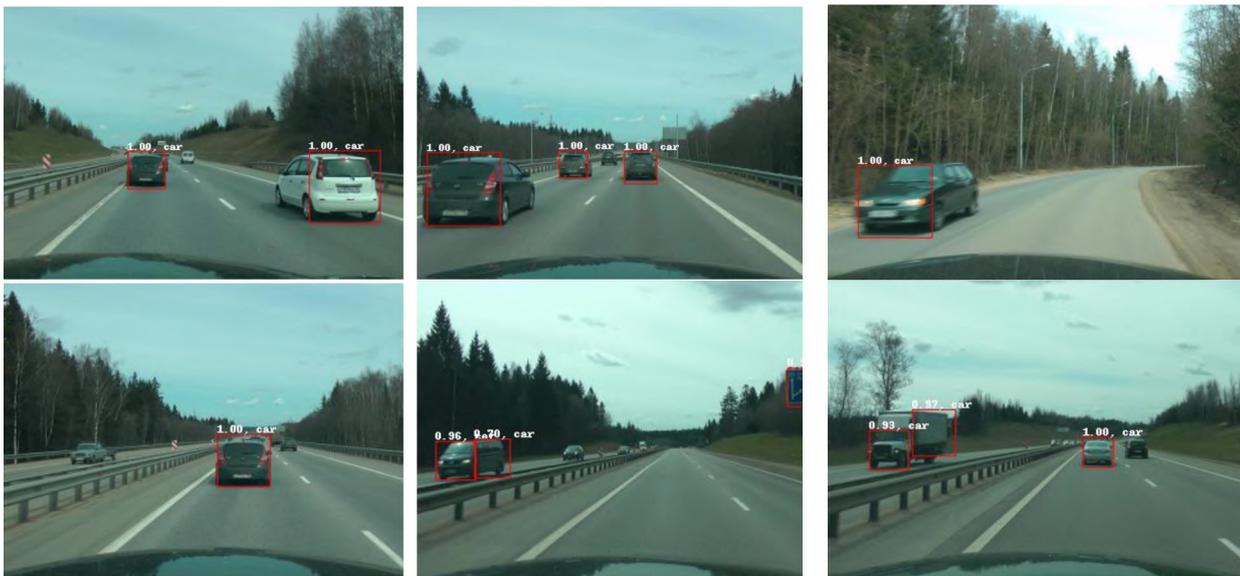


Рис. 4. Примеры обнаружения транспортных средств на изображениях загородных шоссе с помощью сверточной нейронной сети SSD300_VOC_Tuned
Fig. 4. Examples of vehicles on the images of suburban highways using a convolutional neural network SSD300_VOC_Tuned

На рисунке 4 показаны примеры обнаружения транспортных средств на изображениях загородных шоссе с помощью сверточной нейронной сети *SSD300_VOC_Tuned*. На некоторых изображениях видны пропуски обнаружения объектов и ложные срабатывания, причиной этому может быть недостаточно точная разметка эталонных изображений, недостаточное количество обучающей выборки или структура сети.

ЗАКЛЮЧЕНИЕ

В дальнейшей работе над этой темой планируется улучшать результаты работы детектора объектов, подавая на обучение более точно размеченную эталонную выборку, настраивая структуру сети и увеличивая количество изображений с объектами.

Время обработки при распознавании транспортных средств на 7000 изображений составило 200 секунд, обработка одного кадра занимает 28,5 миллисекунд. Такое время обработки одного кадра позволит распознавать потоковое видео с частотой 35 кадров в секунду. Это означает, что с учетом высокого качества обнаружения объектов примененный метод *SSD (Single Shot Multibox Detector)* можно использовать для создания эффективных систем помощи водителю и мониторинга дорожной обстановки.

БЛАГОДАРНОСТИ.

Работа выполнена в рамках гранта Президента РФ для государственной поддержки молодых российских ученых № МК-3130.2017.9 (договор №14.Z56.17.3130-МК).

Список литературы

1. Распознавание транспортных средств и регистрация их траектории движения на последовательности изображений / Юдин Д.А., Горшкова Н.Г., Кныш А.С., Фролов С.В. // Вестник БГТУ им. В.Г. Шухова. 2016. №6. С. 139-148.
2. Cognitive pilot. Система автономного управления наземным транспортом компании «Когнитивные технологии». URL: <http://cognitivepilot.com/ru/about/technologies>.
3. CUDA Toolkit. Develop, Optimize and Deploy GPU-accelerated Apps. URL: <https://developer.nvidia.com/cuda-toolkit>
4. Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, and J. Sun. arXiv preprint arXiv:1512.03385, 2015.
5. Faster r-cnn: Towards real-time object detection with region proposal networks / S. Ren, K. He, R. Girshick, J. Sun // In Advances in neural information processing systems, 2015, P. 91–99.
6. Keras: Deep Learning library for Theano and TensorFlow, URL: <https://keras.io/>.
7. Microsoft COCO: Common objects in context / T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick // In ECCV, 2014.
8. Pickle – Python object serialization. URL: <https://docs.python.org/3/library/pickle.html>
9. Powers, D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies. 2007, 2 (1): 37–63.
10. R-fcn: Object detection via region-based fully convolutional networks / J. Dai, Y. Li, K. He, and J. Sun // arXiv preprint arXiv:1605.06409, 2016.
11. Rykov A. Port of Single Shot MultiBox Detector to Keras. URL: https://github.com/rykov8/ssd_keras.
12. Scalable, high-quality object detection / C. Szegedy, S. Reed, D. Erhan, and D. Anguelov // arXiv preprint arXiv:1412.1441, 2014
13. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
14. Speed/accuracy trade-offs for modern convolutional object detectors / Huang J., Rathod V., Sun C., Zhu M., Korattikara A., Fathi A., Fischer I., Wojna Z., Song Y., Guadarrama S., Murphy K. // arXiv:1611.10012v3 [cs.CV], 2017.
15. SSD: Single Shot MultiBox Detector / Liu W., Anguelov D. and Erhan D. and Szegedy C. and Reed S. and Fu C.-Y. and Berg A. C. // ECCV, arXiv:1512.02325, 2016.

16. TensorFlow. An open-source software library for Machine Intelligence, URL: <https://www.tensorflow.org/>.
17. The Pascal Visual Object Classes (VOC) Challenge / M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. // *International Journal of Computer Vision*, 2010, Vol. 88, Issue 2, P. 303–338.
18. Translation aware fully convolutional instance segmentation / Y. Li, H. Qi, J. Dai, X. Ji, W. Yichen. URL: <https://github.com/daijifeng001/TA-FCN>, 2016.
19. Viola P., Jones M. J. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001, vol. 1, pp. I-511–I-518.
20. You only look once: Unified, real-time object detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // *arXiv preprint arXiv:1506.02640*, 2015.
21. Yudin D., Knysh A. Vehicle recognition and its trajectory registration on the image sequence using deep convolutional neural network // *The International Conference on Information and Digital Technologies*, 2017, P. 435-441.

Reference

1. Vehicle recognition and its trajectory registration on image sequence / Yudin D.A., Gorshkova N.G., Knysh A.S., Frolov S.V. // *Bulletin of BSTU. named after V.G. Shukhov*, vol. 6, pp. 139–148, 2016.
2. Cognitive pilot. Autonomous control system of ground transport by Cognitive Technologies. URL: <http://cognitivepilot.com/ru/about/technologies>.
3. CUDA Toolkit. Develop, Optimize and Deploy GPU-accelerated Apps. URL: <https://developer.nvidia.com/cuda-toolkit>
4. Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, and J. Sun. *arXiv preprint arXiv:1512.03385*, 2015.
5. Faster r-cnn: Towards real-time object detection with region proposal networks / S. Ren, K. He, R. Girshick, J. Sun // *In Advances in neural information processing systems*, 2015, P. 91–99.
6. Keras: Deep Learning library for Theano and TensorFlow, URL: <https://keras.io/>.
7. Microsoft COCO: Common objects in context / T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick // *In ECCV*, 2014.
8. Pickle – Python object serialization. URL: <https://docs.python.org/3/library/pickle.html>
9. Powers, D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*. 2007, 2 (1): 37–63.
10. R-fcn: Object detection via region-based fully convolutional networks / J. Dai, Y. Li, K. He, and J. Sun // *arXiv preprint arXiv:1605.06409*, 2016.
11. Rykov A. Port of Single Shot MultiBox Detector to Keras. URL: https://github.com/rykov8/ssd_keras.
12. Scalable, high-quality object detection / C. Szegedy, S. Reed, D. Erhan, and D. Anguelov // *arXiv preprint arXiv:1412.1441*, 2014.
13. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
14. Speed/accuracy trade-offs for modern convolutional object detectors / Huang J., Rathod V., Sun C., Zhu M., Korattikara A., Fathi A., Fischer I., Wojna Z., Song Y., Guadarrama S., Murphy K. // *arXiv:1611.10012v3 [cs.CV]*, 2017.
15. SSD: Single Shot MultiBox Detector / Liu W., Anguelov D. and Erhan D. and Szegedy C. and Reed S. and Fu C.-Y. and Berg A. C. // *ECCV*, *arXiv:1512.02325*, 2016.
16. TensorFlow. An open-source software library for Machine Intelligence, URL: <https://www.tensorflow.org/>.
17. The Pascal Visual Object Classes (VOC) Challenge / M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. // *International Journal of Computer Vision*, 2010, Vol. 88, Issue 2, P. 303–338.
18. Translation aware fully convolutional instance segmentation / Y. Li, H. Qi, J. Dai, X. Ji, W. Yichen. URL: <https://github.com/daijifeng001/TA-FCN>, 2016.
19. Viola P., Jones M. J. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001, vol. 1, pp. I-511–I-518.
20. You only look once: Unified, real-time object detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // *arXiv preprint arXiv:1506.02640*, 2015.

21. Yudin D., Knysh A. Vehicle recognition and its trajectory registration on the image sequence using deep convolutional neural network // The International Conference on Information and Digital Technologies, 2017, P. 435-441.

Чуйков Роман Юрьевич, магистрант кафедры технической кибернетики

Юдин Дмитрий Александрович, кандидат технических наук, доцент кафедры технической кибернетики

Chuykov Roman Yurievich, master student of the Department of technical cybernetics

Yudin Dmitriy Aleksandrovich, candidate of technical sciences, engineer of the Department of technical cybernetics