

## Нейросетевой анализ транспортных потоков городских агломераций на основе данных публичных камер видеонаблюдения

А. В. Зацерковный<sup>1,a</sup>, Е. А. Нурминский<sup>2,b</sup>

<sup>1</sup>Тихоокеанский океанологический институт им. В. И. Ильичева Дальневосточного отделения РАН,  
Россия, 690041, г. Владивосток, ул. Балтийская, д. 43

<sup>2</sup>Дальневосточный федеральный университет,  
Россия, 690090, г. Владивосток, ул. Суханова, д. 8

E-mail: <sup>a</sup> avz@poi.dvo.ru, <sup>b</sup> nurminskiy.ea@dvfu.ru

Получено 08.12.2020, после доработки — 22.12.2020.

Принято к публикации 15.01.2021.

Адекватное моделирование сложной динамики городских транспортных потоков требует сбора больших объемов данных для определения характера соответствующих моделей и их калибровки. Вместе с тем оборудование специализированных постов наблюдения является весьма затратным мероприятием и не всегда технически возможно. Совокупность этих факторов приводит к недостаточному фактографическому обеспечению как систем оперативного управления транспортными потоками, так и специалистов по транспортному планированию с очевидными последствиями для качества принимаемых решений. В качестве способа обеспечить массовый сбор данных хотя бы для качественного анализа ситуаций достаточно давно применяется обзорные видеокamеры, транслирующие изображения в определенные ситуационные центры, где соответствующие операторы осуществляют контроль и управление процессами. Достаточно много таких обзорных камер предоставляют данные своих наблюдений в общий доступ, что делает их ценным ресурсом для транспортных исследований. Вместе с тем получение количественных данных с таких камер сталкивается с существенными проблемами, относящимися к теории и практике обработки видеоизображений, чему и посвящена данная работа. В работе исследуется практическое применение некоторых мейнстримовских нейросетевых технологий для определения основных характеристик реальных транспортных потоков, наблюдаемых камерами общего доступа, классифицируются возникающие при этом проблемы и предлагаются их решения. Для отслеживания объектов дорожного движения применяются варианты сверточных нейронных сетей, исследуются способы их применения для определения базовых характеристик транспортных потоков. Простые варианты нейронной сети используются для автоматизации при получении обучающих примеров для более глубокой нейронной сети YOLOv4. Сеть YOLOv4 использована для оценки характеристик движения (скорость, плотность потока) для различных направлений с записей камер видеонаблюдения.

Ключевые слова: искусственные нейронные сети, машинное зрение, машинное обучение, сопровождение объекта, сверточные нейронные сети

Исследование выполнено при поддержке Российского фонда фундаментальных исследований (проект 18-29-03071 мк). Исследование Е. А. Нурминского выполнено при поддержке Министерства науки и высшего образования Российской Федерации, дополнительное соглашение 075-02-2020-1482-1 от 21.04.2020.

UDC: 519.8,004.932,519.254,656.13

## Neural network analysis of transportation flows of urban agglomeration using the data from public video cameras

A. V. Zatserkovnyy<sup>1,a</sup>, E. A. Nurminski<sup>2,b</sup>

<sup>1</sup>Pacific oceanological institute Far eastern branch of RAS,  
43 Baltiyskaya st., Vladivostok, 690041, Russia

<sup>2</sup>Far Eastern Federal University,  
8 Sukhanov st., Vladivostok, 690090, Russia

E-mail: <sup>a</sup> avz@poi.dvo.ru, <sup>b</sup> nurminskiy.ea@dvfu.ru

*Received 08.12.2020, after completion — 22.12.2020.*

*Accepted for publication 15.01.2021.*

Correct modeling of complex dynamics of urban transportation flows requires the collection of large volumes of empirical data to specify types of the modes and their identification. At the same time, setting a large number of observation posts is expensive and technically not always feasible. All this results in insufficient factographic support for the traffic control systems as well as for urban planners with the obvious consequences for the quality of their decisions. As one of the means to provide large-scale data collection at least for the qualitative situation analysis, the wide-area video cameras are used in different situation centers. There they are analyzed by human operators who are responsible for observation and control. Some video cameras provided their videos for common access, which makes them a valuable resource for transportation studies. However, there are significant problems with getting qualitative data from such cameras, which relate to the theory and practice of image processing. This study is devoted to the practical application of certain mainstream neuro-networking technologies for the estimation of essential characteristics of actual transportation flows. The problems arising in processing these data are analyzed, and their solutions are suggested. The convolution neural networks are used for tracking, and the methods for obtaining basic parameters of transportation flows from these observations are studied. The simplified neural networks are used for the preparation of training sets for the deep learning neural network YOLOv4 which is later used for the estimation of speed and density of automobile flows.

Keywords: artificial neural networks, computer vision, machine learning, object tracking, convolutional neural networks, YOLO

Citation: *Computer Research and Modeling*, 2021, vol. 13, no. 2, pp. 305–318 (Russian).

This work was supported by the Russian Foundation for Basic Research, grant 18-29-03071 mk. The work by E. A. Nurminski was supported by Ministry of Science and Higher Education of the Russian Federation, supplementary agreement 075-02-2020-1482-1 from 21.04.2020.

## Введение

Одной из проблем компьютерного моделирования реальных транспортных потоков городских агломераций в условиях Российской Федерации является их недостаточное обеспечение экспериментальными данными по основным параметрам этих потоков: скорость, плотность, распределение по типам транспортных средств (ТС) и пр. Даже в крупных городах лишь немногие из транспортных артерий оборудованы соответствующими датчиками, наиболее эффективными из которых являются индуктивные, работа которых не зависит от погодных условий, освещения и которые не требуют установки на дороге дополнительных сооружений. Вместе с тем такие датчики отличаются довольно высокой стоимостью, существует необходимость проведения строительных работ для установки под полотном дороги, обеспечения питанием, защитой от стихий, системами передачи данных и многим другим, что существенно удорожает их установку и эксплуатацию и затрудняет модификацию в связи с постоянно изменяющимися дорожными условиями в городах. Наибольшее распространение в России имеют радиолокационные дорожные датчики, которые, однако, имеют ограниченные возможности по распознаванию типов транспортных средств, в связи с чем их приходится комбинировать с другими устройствами типа ультразвуковых и инфракрасных детекторов, что создает дополнительные проблемы синхронизации измерений. Наиболее перспективными считаются измерительные системы, использующие видеодетекторы, хотя качество их информации сильно зависит от погодных условий. Дополнительным преимуществом видеосистем мониторинга трафика является широкое распространение публично доступных систем видеонаблюдений, достаточно популярных у современных телекоммуникационных провайдеров. В частности, например, во Владивостоке один ИСП «Подряд» имеет порядка 60 камер наблюдения за дорожной обстановкой, «Альянс Телеком» — около 30 камер, «VladLink» — 11 и т. д. Видеопотоки с этих камер содержат большой объем информации о наблюдаемом транспортном потоке и при соответствующей обработке с минимальными затратами и максимальной оперативностью могут предоставить необходимые для транспортных исследований калибровочные параметры фундаментальных диаграмм, вероятностные характеристики потоков, динамику их изменения во времени и многое другое.

Тема анализа дорожных видеоизображений для различных задач транспортных исследований рассматривалась фактически с самого их начала: пионерская работа инженера Bruce D. Greenshields, которая внесла в транспортную науку понятие фундаментальной диаграммы, была основана на обработке видеоизображений движущихся автомобилей, зафиксированных на киноплёнке [Fundamental Diagram for Traffic, 2011]. Современные технологии сделали в этом направлении гигантский шаг вперед, выведя эти технологии на уровень массового практического применения для мониторинга дорожных условий и управления движением. Вместе с тем представляется, что потенциал видеонаблюдений в этой области использован еще не в полной мере, что в немалой степени связано с недостаточным развитием информационных технологий их обработки и ассимиляции этой информации в системы моделирования и управления движением. В то время как средства анализа и классификации изображений (см., например, [Image Processing Tools]) уже достаточно хорошо себя зарекомендовали при анализе статических данных, видеонаблюдение за дорожным трафиком требует обработки не всегда отчетливых изображений движущихся тел в изменяющихся условиях, что представляет собой определенную проблему. Направление работ по нахождению и классификации движущихся объектов в видеопотоке называют VOT (Visual Object Tracking); с основными подходами в этой области можно ознакомиться в работе [Lu, Wang, 2019].

## Нейросетевой анализ транспортных потоков городских агломераций

Мировая практика в области использования видеоизображений для анализа автомобильного трафика сориентировалась на использование высококачественных видеопотоков, полученных со специальной аппаратуры. Достаточно скромные ресурсы большинства отечественных исследователей и практиков в этой области вынуждают искать бюджетные варианты получения адекватных данных транспортной статистики, и одним из популярных вариантов является как раз использование общедоступных видеокамер общего вида. Вместе с тем низкое качество изображений, полученных с общедоступных камер, расположенных, как правило, далеко от перекрестков и транспортных артерий (обычной практикой являются крыши высоких зданий поблизости), вынуждает развивать передовые системы и алгоритмы обработки данных, основанные, в частности, на развитых нейросетевых технологиях [Нурминский, 2014; Khazukov et al., 2020; Fedorov et al., 2019]. Широкому применению нейронных сетей способствует появление развитых открытых платформ разработки (Tensorflow/Keras/PyTorch/MXnet/..., [Tensorflow; Keras; Pytorch; MXnet]), доступных образовательных инициатив (Deeplearning.AI/Kaggle/Coursera/EDX/D2L.AI..., [Deeplearning.AI; Kaggle; Coursera MOOC; EDX MOOC; Zhang et al., 2021]), открытых и публично доступных реализаций нейронных сетей, решающих конкретные прикладные задачи, в частности определение объектов различных классов на изображении. Например, в публичном доступе есть обученные модели, которые позволяют локализовать и классифицировать объекты дорожного движения в довольно широком диапазоне условий [Tensorflow hub models; YOLO model]. Недавно было объявлено [Bochkovskiy et al., 2020] об очередном существенном повышении эффективности этих нейросетей.

### Получение первичных данных и обучение нейронных сетей

С точки зрения обработки видеопоток можно рассматривать как последовательность изображений (кадров или фреймов), получаемых светочувствительным сенсором камеры (фотоматрицей) через объектив с некоторой, возможно переменной, частотой (частота кадров или FPS). Черно-белое изображение представляет собой матрицу целых значений (пиксель), каждое из которых описывает интенсивность светового потока, попавшего в определенную ячейку сенсора, значениями от 0 до 255, если глубина цвета 1 байт либо в большем диапазоне, если использовано представление глубиной 1.5, 2 или 4 байта. Размерность матрицы называют разрешением изображения. Популярное для современных видеокамер разрешение FullHD предполагает матрицу  $1920 \times 1080$  пикселей, в набирающем популярность разрешении 4k UHD используется матрица  $3840 \times 2160$  пикселей.

Для представления цветного изображения каждый пиксель описывается группой значений. Например, в популярной модели представления цвета RGB (Red, Green, Blue) пиксель описывается тремя значениями, каждое из которых может варьироваться от 0 до 255, если используется глубина 1 байт, либо в большем диапазоне. При использовании глубины каждого цвета 1 байт цветное RGB-изображение можно рассматривать как матрицу из трехкомпонентных однобайтовых векторов либо как три матрицы с однобайтовыми целыми. Такое изображение может принимать  $2^{24 \times 1920 \times 1080}$  вариантов значений.

Несжатое FullHD RGB-изображение занимает больше 6 Мбайт. Одна секунда полного цветного видеопотока, при частоте кадров 25 фреймов в секунду, занимает больше 155 Мбайт. К счастью, последовательность кадров видео обычно содержит много повторяющейся информации, что используется для сжатия потока. Чаще используются алгоритмы, теряющие некоторую часть информации об оригинальных фреймах. Сжатый поток с камеры может полностью либо выборочно записываться в систему хранения в виде последовательности файлов, часто

с перезаписью поверх устаревших данных. Объем сжатого видео обычно зависит от динамики сцены. Статичное видео обычно сжимается лучше.

Доступные модели нейронных сетей тренируют и тестируют на публичных наборах данных [Coco Dataset; PASCAL VOC dataset; Imagenet VOT dataset; VOT Chalange dataset; YOLO VOC dataset; Kaggle] либо на данных, собранных для конкретной задачи. Часто берется первый вариант и доучивается для конкретной задачи на данных, отражающих специфику приложения. Последние слои модели можно адаптировать к задаче, например, избавившись от лишних классов.

Набор для обучения предполагает информацию, которая подается на вход нейронной сети (например, множество изображений), и метаданные о входных данных, которые используются для вычисления функции ошибки. Функция ошибки (loss function) выступает в роли целевой функции для оптимизации коэффициентов модели. В качестве метаданных могут выступать, например, описания класса объекта в каждом изображении (задача классификации изображения) или наличие, класс, положение и размеры объектов (задача определения, классификации и локализации объектов в изображении).

Учитывая разнообразие условий получения видеопотоков наблюдения за дорожным движением, выбранная с публичных репозиториев модель нейронной сети может потребовать дополнительное обучение для адаптации коэффициентов модели к конкретным условиям съемки. Например, для приложения, описанного в этой работе, модель YOLOv3 [YOLO model], обученная для оценки дорожной ситуации с автомобильной камеры (эта модель используется в обучающем примере в популярном курсе [DeepLearning.AI CNN course]), ненадежно определяла контуры автомобилей на большом расстоянии от стационарной камеры, расположенной на крыше здания (см. рис. 1).

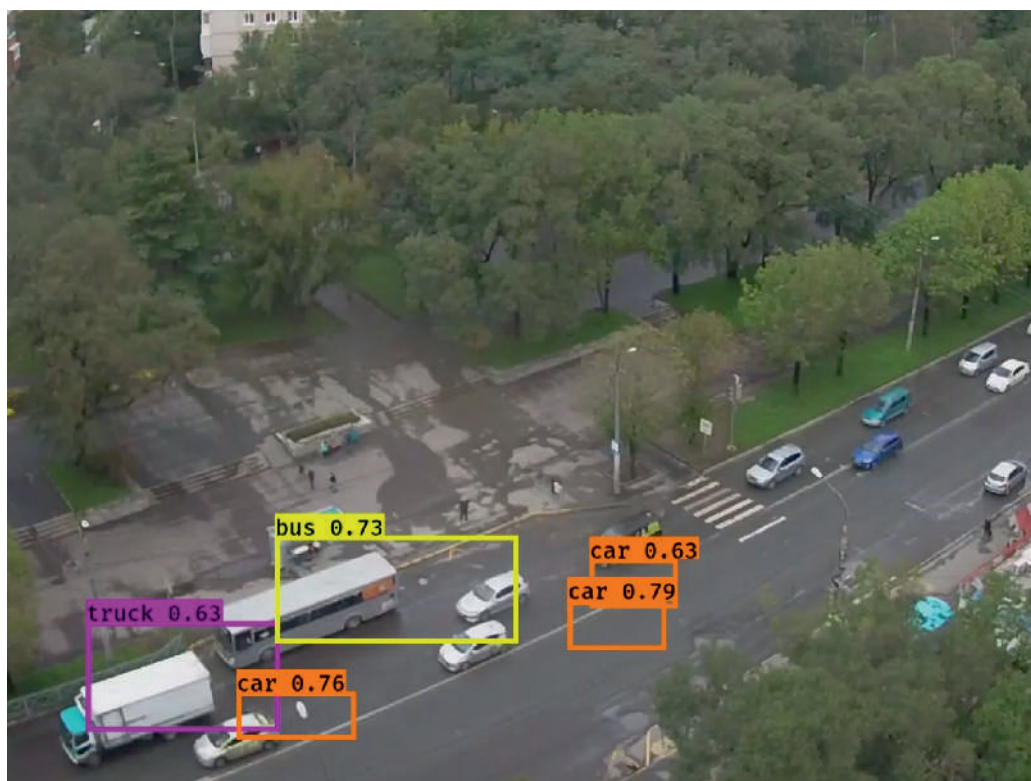


Рис. 1. Оценка непереученной модели YOLOv3 на данных одной из публичных дорожных камер Владивостока

Модель оценивала вероятность наличия объекта в каждой из множества ячеек, покрывающих изображение, локализуя прямоугольный бокс объекта, его положение, размеры и вероятность принадлежности объекта классам. В приведенном примере определены 5 боксов с наибольшими значениями оценки вероятности наличия объекта. Типы объектов определены точно, положение боксов — с заметной ошибкой. Для транспортных моделей это существенно, так как влияет на оценки плотности дорожного потока в полосах движения и скорости движения автомобилей.

Один из основных путей для повышения качества распознавания либо локализации нейронной сети, обученной на сторонних данных, — дополнительная тренировка сети на собственных данных. Ручная подготовка метаданных для обучения является довольно трудоемким процессом, поэтому их получение по видеопотоку или отдельным изображениям представляет собой самостоятельную проблему. Требуемый объем примеров зависит от сложности задачи и от того, насколько анализируемые моделью изображения и объекты отличаются от примеров, на которых модель тренировали. Эта разница не всегда очевидна, если учесть обычную практику получения множества дополнительных примеров путем различных преобразований оригинального изображения (с автоматическим изменением метаданных, если требуется). Если примеров достаточно много, иногда эффективнее обучить модель с нуля. В моделях промышленного качества надежность достигается за счет тренировки на миллионах подходящих к задаче примеров.

В данном случае автоматизация получения метаданных достигалась средствами библиотеки OpenCV для определения контуров объектов-автомобилей (см. [OpenCV bg. subtraction; Contours in OpenCV; Kaehler, Bradski, 2016; Forsyth, Ponce, 2012]) и средствами простой нейронной сети для классификации содержимого контуров. Пример результатов автоматического оконтуривания приведен на рис. 2, и этот пример демонстрирует также часто встречающуюся в плотных потоках ситуацию, когда два контура сливаются в один (второе изображение слева).

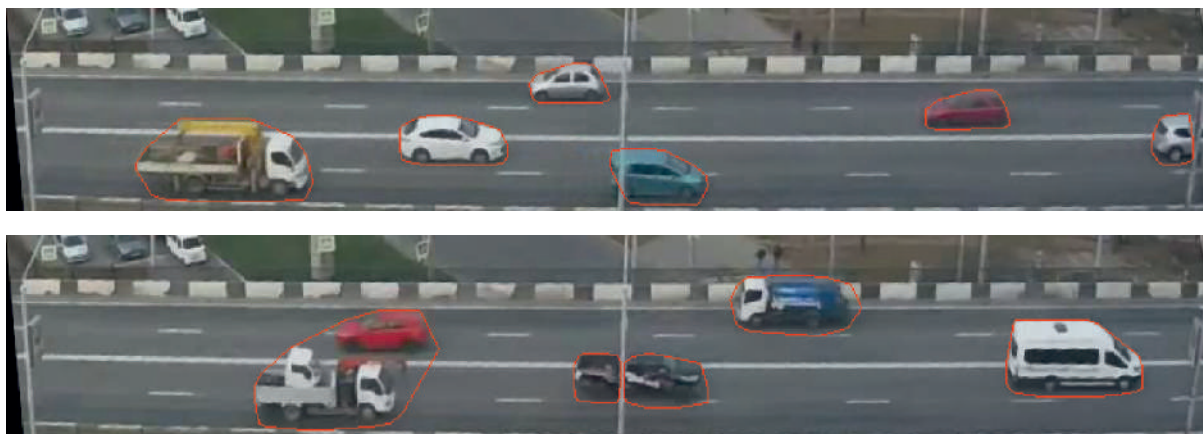


Рис. 2. Автоматически оконтуренные средствами OpenCV изображения движущихся автомобилей

Данные о контурах автомобилей используются для оценки их особенностей, для тренировки нейронных сетей и для оценки работы моделей. Даже если готовая доступная CNN/RNN дает результаты, которые выглядят надежными, источник верифицированных местных данных позволяет провести оценку точности модели в требуемых условиях.

Последовательность видеофреймов содержит множество изображений проезжающего авто, поэтому имеет смысл собрать эти изображения в группу (назовем ее «след конкретного авто»). OpenCV содержит стандартные функции, которые отслеживают эволюцию содержимого контура в последовательных фреймах, однако в данной работе использовалась собственная реализация сопровождения контура.

Каждый контур в видео можно идентифицировать парой: номер фрейма и уникальный идентификатор контура в данном фрейме (например, номер в списке контуров данного фрейма). Множество следов в конкретном видео можно представить в виде словаря, ключом в котором является идентификатор последнего контура следа.

- 1: На каждом временном шаге контуры, обнаруженные в следующем фрейме, проверяются на принадлежность к одному из актуальных для предыдущего фрейма следов по площади пересечения с контуром, отнесенным к следу в предыдущем фрейме.
- 2: Если для некоторого контура не обнаружена принадлежность к существующему следу, начинаем новый след.

След, во-первых, позволяет упростить ручную классификацию множества контуров и, во-вторых, предоставляет информацию о движении конкретных автомобилей. След в таком же виде можно использовать и с контурами (боксами), полученными от нейронной сети.

## Автоматизация классификации обучающих данных

Следующая существенная проблема состоит в определении типов автомобилей, однако в городских условиях Владивостока более 90 % популяции транспортных средств составляют легковые автомобили. Соответственно, та же пропорция легковых автомобилей будет доминировать и в обучающей выборке, полученной по реальным наблюдениям, и качество распознавания других типов транспортных средств, отличных от легковых автомобилей, заведомо будет невысоким из-за малого числа их примеров.

Чтобы обработать больше примеров и классифицировать вручную только следы с относительно редкими видами объектов, сначала была автоматизирована классификация легковых авто.

Для бинарной классификации «легковой автомобиль или нет» содержимого контуров было решено использовать последовательную CNN простой архитектуры, описанную в таблице 1.

Таблица 1. Описание последовательной сверточной нейронной сети (CNN), использованной для бинарной классификации содержимого контуров примеров. Модель содержит 4 321 089 тренируемых параметров

Имя уровня	Описание	Размерность на выходе	Количество параметров
Rescaling	Масштабирование, нормализация ввода	$49 \times 85 \times 3$	0
Conv2D-1	Сверточный уровень, активация Relu, ядра $3 \times 3$ , 32 фильтра	$49 \times 85 \times 32$	846
MaxPooling2D-1	Полинг $2 \times 2$ по максимуму	$24 \times 42 \times 32$	0
Conv2D-2	Сверточный уровень, активация Relu, ядра $3 \times 3$ , 64 фильтра	$24 \times 42 \times 64$	18 496
MaxPooling2D-2	Полинг $2 \times 2$ по максимуму	$12 \times 21 \times 64$	0
Conv2D-3	Сверточный уровень, активация Relu, ядра $3 \times 3$ , 128 фильтров	$12 \times 21 \times 128$	73 856
MaxPooling2D-3	Полинг $2 \times 2$ по максимуму	$6 \times 10 \times 128$	0
Conv2D-4	Сверточный уровень, активация Relu, ядра $3 \times 3$ , 256 фильтров	$6 \times 10 \times 256$	295 168
Dropout	Регуляризация dropout, $p = 0.1$	$6 \times 10 \times 256$	0
Flatten	Превращение тензора в одномерный массив	15 360	0
Dense-1	Полносвязный уровень, активация Relu	256	3 932 416
Dense-2	Полносвязный уровень, активация Sigmoid	1	257

Для обучения этой сети использовалась небольшая выборка примерно из 4000 следов легковых машин (это порядка 120 000 контуров) и около 1000 следов других типов (30 000 контуров). Для случая рассматриваемой камеры 99.5 % контуров легковых машин помещались в прямоугольник  $49 \times 85$  пикселей (высота, ширина).

Сравниваемый образец можно растянуть либо дополнить до указанного размера, если пример изображения меньше  $49 \times 85$ , либо сжать или вырезать часть, если пример больше. Недостаточное количество контрпримеров можно компенсировать за счет больших размеров контуров значительной части грузовиков и автобусов. Из прямоугольника большего размера можно нарезать несколько образцов размером  $49 \times 85$ , выбирая их положение случайным образом. Как мы отмечали, описание следов содержит номера пересекаемых ими фреймов. Это дает возможность вычислить фреймы, которые не содержат контуры (т. е. фреймы с чистым от движения фоном). Также используем эти фреймы для получения контрпримеров, блуждая по изображению фрейма рамкой  $49 \times 85$ .

От сети требуется точность, которую можно увеличивать, постепенно дотренировывая с ростом количества классифицированных образцов. Верификация результатов на первых этапах выполняется вручную, с увеличением объема примеров неизбежен переход к проверке результатов классификации статистических выборок образцов.

Обученная модель показала хорошие результаты на контурах из контрольного видео, которое не участвовало в составлении обучающего набора данных, при этом достигалась точность 99.67 % на легковых автомобилях и 98.9 % на альтернативах.

На рис. 3 в строчке Predicts распечатаны результаты модели на соответствующем изображении. Ситуация по первому следу очевидна. Некоторые колебания модели по первому неполному изображению грузовика можно понять. Второй пример интереснее. Модель пока не обучалась легковым авто с прицепами, поэтому у данного следа установлен тип «прочие». Пока прицеп не виден (10 боксов, на 11-м прицеп едва заметен), модель уверенно узнает легковое авто, но по мере появления в боксе прицепа сомнения усиливаются, и далее боксы не признаются легковым авто. Последний пример может показаться загадочным (1 и 13 боксы уверенно указаны как не легковое авто), но, если присмотреться, у 1 и 13 в контур попало много фона, и, поскольку для решения задачи нужны образцы с точным оконтуриванием, модель помогает браковать такие примеры.

Модель позволяет классифицировать вручную только те следы, которые дали относительно низкую вероятность принадлежности к легковым авто (скажем, меньше 0.8), а это уменьшает количество просматриваемых и классифицируемых вручную следов многократно. Так, можно просматривать большее количество следов, формируя достаточную базу для автоматической типизации по всем классам.

Естественно, что после анализа ошибок сети и многократного доучивания остается вероятность ошибок классификации, поэтому не следует забывать о периодических проверках выборок образцов как на соответствие типам, так и на точность оконтуривания.

При классификации примера сеть возвращает значение между нулем и единицей, которое можно интерпретировать как оценку вероятности принадлежности к классу. Эту же сеть, почти без изменений, можно использовать для другой задачи — для определения положения границы тени автомобиля внутри контура. Если для бинарной классификации в метаданных к обучающему примеру указывается 1 для изображений с легковым авто и 0 для прочих изображений, то для определения границы в метаданных обучающих примеров указывается относительное положение границы тени внутри бокса (число между 0 и 1). Следующий шаг развития в этом направлении — сеть, которая возвращает два значения: оценку принадлежности изображения к легковым авто и относительное положение тени. Сеть может иметь ту же архитектуру, но на последнем слое она должна вернуть два значения. Метаданные примеров должны содержать







Рис. 4. Результаты работы сети YOLOv4 на изображениях наблюдаемой зоны движения в различных условиях

Данная модель гораздо сложнее в сравнении с приведенной в начале параграфа (было 4 слоя Conv2D, 4.3 миллиона коэффициентов). Используемая сеть YOLOv4 содержит более 64.4 миллиона коэффициентов, 110 слоев Conv2D. Естественно, оценка видеопотока с помощью такого мощного инструмента требует значительных вычислительных ресурсов. На четырехъядерном процессоре i5-8300H 30 секундный ролик с разрешением  $764 \times 128$  с частотой кадров 25 FPS (это 750 фреймов) анализировался около 10 минут при полной загрузке всех ядер. Специализированный сервер с GPU Tesla V100-PCI-E-32GB выполняет анализ всех фреймов 8-минутного ролика (это 12 000 фреймов) за 7–9 минут. То есть на специализированном для AI GPU анализ изображений данной сетью выполняется примерно в 15 раз быстрее. Отметим, что тренировка сетей, при правильно подобранном размере пакета примеров (mini batch), может выполняться на GPU на два порядка быстрее, чем на CPU.

## Экспериментальная оценка дорожного движения

Проверка возможностей системы отслеживания движения автомобилей с CNN YOLOv4 проведена для конкретной зоны движения (центр путепровода на проспекте 100-летия Владивостока над улицей Русской) для двух направлений: в сторону центра Владивостока и в сторону выезда из города. Учитывая положение зоны внимания, для оценки скоростей можно использовать разность горизонтальных координат границ первого и последнего боксов, образующих след на некотором множестве фреймов.

Разделив эту разность на количество фреймов, в которых существует след, можно получить оценку скорости в пикселях на фрейм. Для пересчета этой скорости в километры в секунду можно, например, подсчитать количество пикселей между фонарями, которые видны на изображениях зоны (около 371 пикселя), и учесть, что расстояние между фонарями составляет около 25 м (оценку в метрах можно проверить с помощью онлайн-карт либо с помощью дальномера). Таким образом вычисляется коэффициент  $25/(371 \times 1000)$  для пересчета пикселей в километры.

Учитывая, что данная камера дает фиксированные 25 фреймов в секунду, вычисляется коэффициент для пересчета фреймов в часы  $1/(25 \times 3600)$ .

Для анализа трафика за неделю была организована периодическая запись роликов с данной камеры. 8-минутные ролики записывались каждые 20 минут в интервале времени с 9:30 до 19:10 с 7 по 13 декабря. В ходе эксперимента было собрано 192 ролика. Следы вычислялись по множеству боксов, полученных на каждом фрейме сетью YOLOv4. Описания боксов (координаты углов, класс, оценка вероятности), описание следов со ссылками на входящие в след боксы, номера фреймов, в которых не обнаружено боксов (образцы фона), вычислялись для каждого ролика и сохранялись в отдельный файл hdf5. По этим данным строились таблицы с оценками скоростей трэйсов, количеством проездов автомобилей для каждого из направлений движения, количеством боксов на фрейм для каждого из направлений.

## Транспортные диаграммы

После обработки порядка 2.5 млн фреймов с видеоизображениями дорожных ситуаций представилась возможность попытаться описать зависимости между основными характеристиками транспортных потоков: скорость, плотность и объем проходящего трафика. Некоторые из этих обобщенных зависимостей, известные как фундаментальные диаграммы, представлены на рис. 5, 6, 7 и 8.

Из анализа рис. 5 можно сделать предварительный вывод о том, что на городское движение существенное влияние оказывает ограничение скорости: до тех пор, пока плотность движения, по сути дела, позволяла бы двигаться быстрее, скорость потока оставалась постоянной, в пределах разрешенного лимита. В определенной степени это предоставляет эмпирическое обоснование модели стационарной динамики [Nesterov, 2000; Nesterov, de Palma, 2000], однако и указывает на необходимое усложнение моделей городского трафика, которые должны учесть явное наличие скачкообразного уменьшения скорости в районе критической плотности, когда фактором уменьшения скорости становится плотность потока, а не регуляторное ограничение. Можно надеяться, что дальнейшие наблюдения позволят уточнить детали подобной фундаментальной диаграммы.

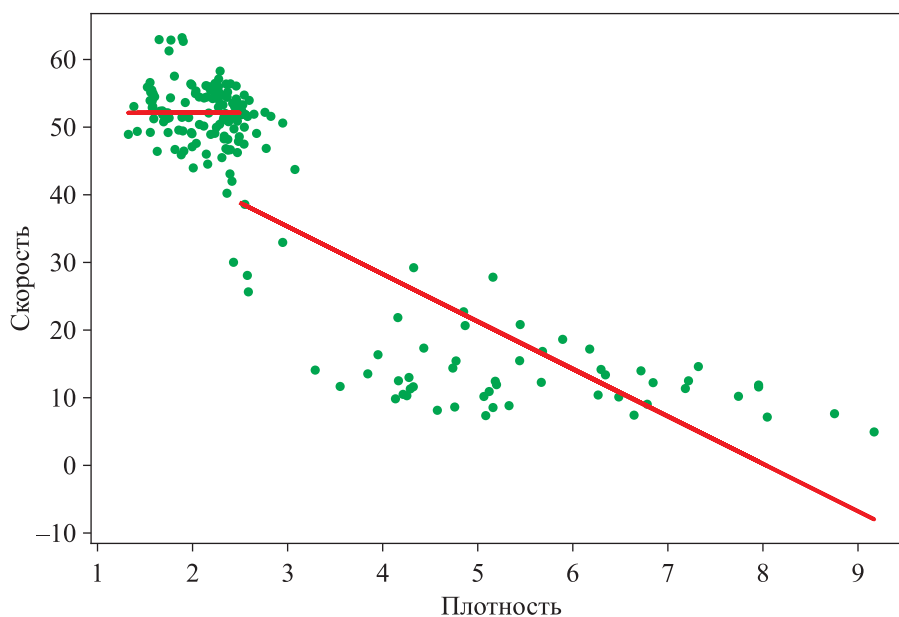


Рис. 5. Соотношение «скорость–плотность» для направления в центр города

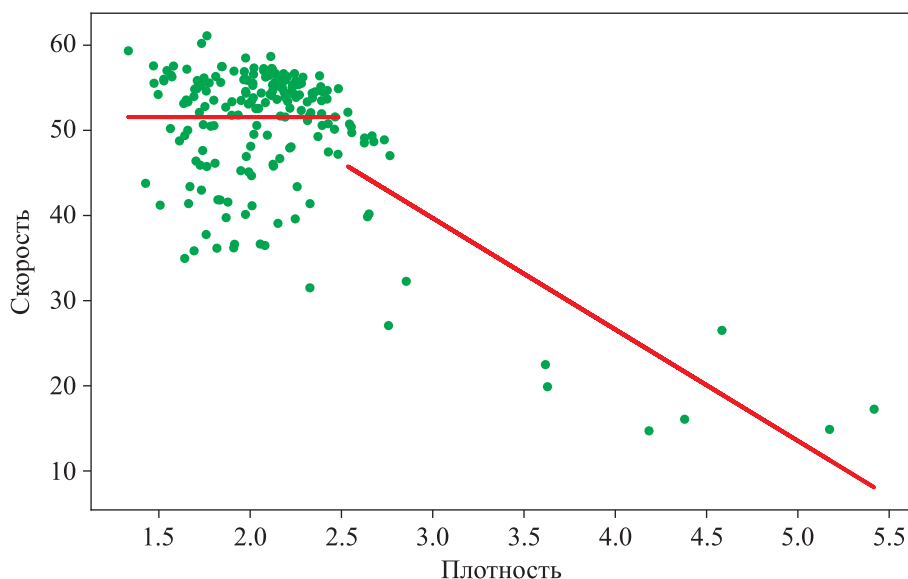


Рис. 6. Соотношение «скорость – плотность» для направления из центра города

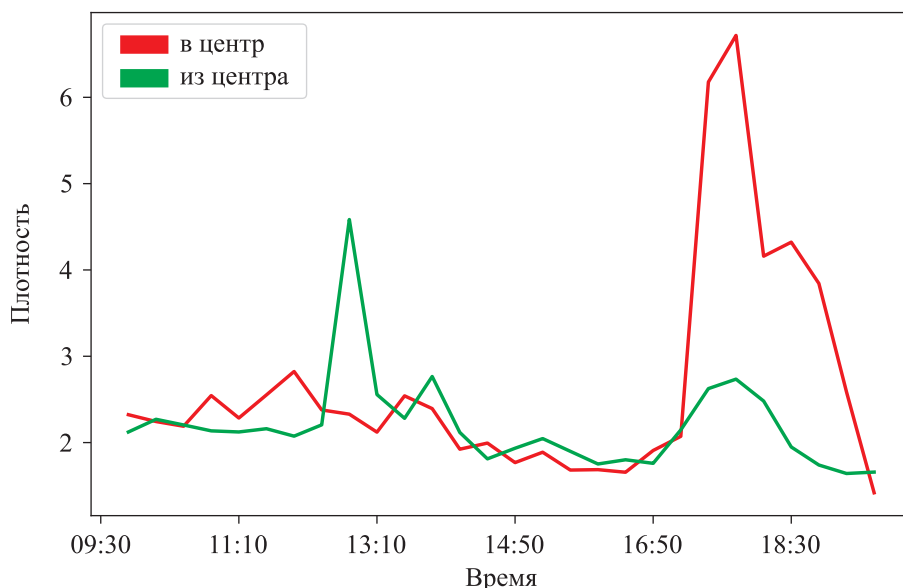


Рис. 7. Динамика плотностей потоков в рабочие часы

При плотности больше некоторой критической скорость уже начинает замедляться в принципе по закону, близкому к классическому линейному. Второй интересный вывод, который можно сделать из рис. 7, заключается в несимметричности направлений, что выглядит странно с точки зрения маятниковой миграции: поездка в обратном направлении не является сменой полос движения, тут работают какие-то иные правила маршрутизации.

## Выводы

По результатам нашей работы можно сделать вывод о том, что YOLOv4 — мощная и гибкая модель, хорошо показавшая себя для задачи анализа дорожного видео. Вместе с тем при ее использовании онлайн-анализ видео, особенно высокого разрешения, требует больших вычислительных затрат. Вероятно, для специализированных задач есть смысл пробовать менее глубокие

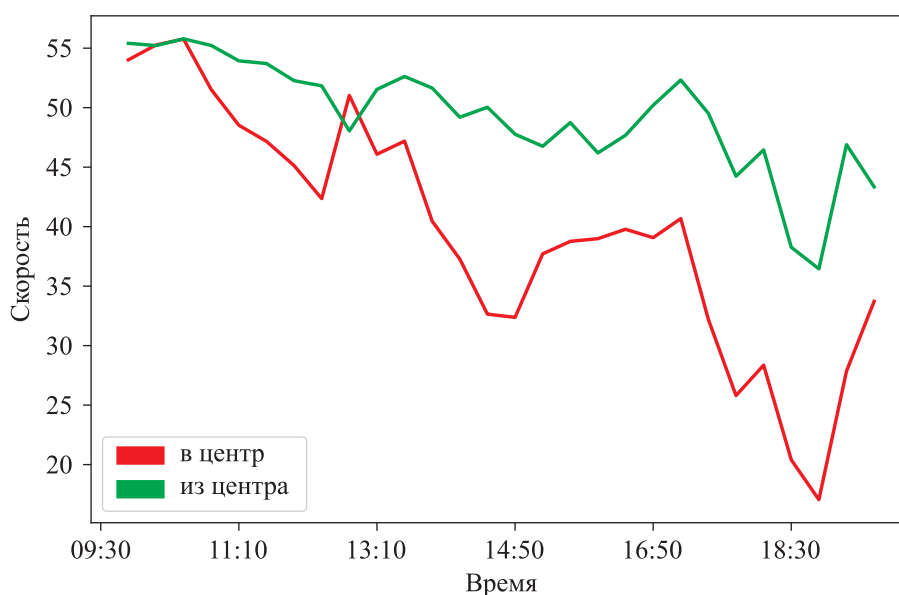


Рис. 8. Динамика скоростей потоков в рабочие часы

варианты YOLO либо адаптированные альтернативы (см. [Tensorflow hub models]). С YOLO предлагается вариант *tiny yolo*, в котором всего несколько слоев, но этот вариант слишком облегчен и не обеспечивает желаемой точности. Из результатов наблюдений можно также извлечь определенные рекомендации для разработчиков транспортных моделей.

## Список литературы (References)

- Нурминский Е. А., Федосеев А. А., Тормозов В. С. Автоматическое определение плотности автомобильного потока по данным камер видеонаблюдения // Информатика и системы управления. — 2014. — Вып. 1. — С. 151–159.
- Nurminskii E. A., Fedoseev A. A., Tormozov V. S. Avtomaticheskoe opredelenie plotnosti avtomobil'nogo potoka po dannym kamer videonablyudeniya [Automatic determination of the density of car traffic according to the data of CCTV cameras] // Informatika i sistemy upravleniya [Computer science and control systems]. — 2014. — Iss. 1. — P. 151–159 (in Russian).
- 75 Years of the Fundamental Diagram for Traffic Flow Theory: Greenshields Symposium // TRB Transportation Research Electronic Circular E-C149, 2011 [Electronic resource]. — <http://onlinepubs.trb.org/onlinepubs/circulars/ec149.pdf> (accessed: 15.02.2021).
- Best Image Processing Tools Used in Machine Learning [Electronic resource]. — <https://neptune.ai/blog/best-image-processing-tools-used-in-machine-learning> (accessed: 15.02.2021).
- Bochkovskiy A., Wang C.-Y., Liao H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection arXiv:2004.10934 [cs.CV].
- Coco Dataset [Electronic resource]. — <https://cocodataset.org> (accessed: 15.02.2021).
- Contours in OpenCV tutorial [Electronic resource]. — [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_contours/py\\_table\\_of\\_contents\\_contours/py\\_table\\_of\\_contents\\_contours.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_table_of_contents_contours/py_table_of_contents_contours.html) (accessed: 15.02.2021).
- Coursera, massive open online course provider [Electronic resource]. — <https://www.coursera.org> (accessed: 15.02.2021).
- Datondji R., Dupuis Y., Subirats P., Vasseur P. A Survey of Vision-Based Traffic Monitoring of Road Intersections // April 2016, IEEE Transactions on Intelligent Transportation Systems. — DOI: 10.1109/TITS.2016.2530146

- DeepLearning.AI Convolutional Neural Networks, Coursera course [Electronic resource]. — <https://www.coursera.org/learn/convolutional-neural-networks> (accessed: 15.02.2021).
- DeepLearning.AI Machine Learning and Deep Learning Specializations and courses [Electronic resource]. — <https://www.deeplearning.ai> (accessed: 15.02.2021).
- EDX, massive open online course provider [Electronic resource]. — <https://www.edx.org> (accessed: 15.02.2021).
- Fedorov A., Nikolskaia K., Ivanov S. et al.* Traffic flow estimation with data from a video surveillance camera // *J Big Data*. — 2019. — Vol. 6, No. 73. — <https://doi.org/10.1186/s40537-019-0234-z>
- Forsyth D. A., Ponce J.* Computer Vision: A Modern Approach. — Pearson, 2ed, 2012.
- Imagenet VOT dataset [Electronic resource]. — <http://image-net.org/download-bboxes> (accessed: 15.02.2021).
- Lu H., Wang D.* Online Visual Tracking. — Springer, 2019.
- Kaehler A., Bradski G.* Learning OpenCV 3. — O'Reilly Media, 2016.
- Kaggle, online community of data scientists [Electronic resource]. — <https://www.kaggle.com> (accessed: 15.02.2021).
- Keras, popular deep learning API [Electronic resource]. — <https://keras.io> (accessed: 15.02.2021).
- Khazukov K., Shepelev V., Karpeta T. et al.* Real-time monitoring of traffic parameters // *J Big Data*. — 2020. — Vol. 7, No. 84. — <https://doi.org/10.1186/s40537-020-00358-x>
- MXnet, open source deep learning framework [Electronic resource]. — <https://mxnet.apache.org> (accessed: 15.02.2021).
- Nesterov Yu.* Stable Traffic Equilibria: Properties and Applications // *Optimization and Engineering*. — 2000. — Vol. 3. — P. 29–50.
- Nesterov Yu., de Palma A.* Stable Dynamics in Transportation Systems // CORE DP 2000/27. — 2000. — Belgium: Catholic University of Louvain.
- OpenCV background subtraction tutorial [Electronic resource]. — [https://docs.opencv.org/4.5.0/d1/dc5/tutorial\\_background\\_subtraction.html](https://docs.opencv.org/4.5.0/d1/dc5/tutorial_background_subtraction.html) (accessed: 15.02.2021).
- PASCAL VOC dataset [Electronic resource]. — <https://pjreddie.com/projects/pascal-voc-dataset-mirror> (accessed: 15.02.2021).
- Pytorch, open source machine learning framework [Electronic resource]. — <https://pytorch.org> (accessed: 15.02.2021).
- Tensorflow, end-to-end open source platform for machine learning [Electronic resource]. — <https://www.tensorflow.org> (accessed: 15.02.2021).
- Tensorflow hub object detection models [Electronic resource]. — <https://tfhub.dev/s?module-type=image-object-detection> (accessed: 15.02.2021).
- VOT Chalange dataset [Electronic resource]. — <https://www.votchallenge.net/vot2020/dataset.html> (accessed: 15.02.2021).
- YOLO object detection model [Electronic resource]. — <https://pjreddie.com/darknet/yolo> (accessed: 15.02.2021).
- YOLO VOC dataset mirror [Electronic resource]. — <https://pjreddie.com/projects/pascal-voc-dataset-mirror> (accessed: 15.02.2021).
- Zhang A., Lipton Z. C., Li M., Smola A. J.* Dive into Deep Learning interactive book [Electronic resource]. — <https://d2l.ai> (accessed: 15.02.2021).