

ОБНАРУЖЕНИЕ ПЕШЕХОДОВ НА ИЗОБРАЖЕНИЯХ НА ОСНОВЕ МОДЕЛИ ГЛУБОКОГО ОБУЧЕНИЯ

М.М. Ляшева, С.А. Ляшева, М.П. Шлеймович

Казанский национальный исследовательский технический университет
им. А.Н. Туполева-КАИ
Российская Федерация, 420111, г. Казань, ул. К. Маркса, д. 10

Аннотация. В работе рассмотрена задача обнаружения пешеходов на изображениях в интеллектуальных системах поддержки водителей транспортных средств. Решение указанной задачи имеет важное значение при разработке и внедрении интеллектуальных транспортных систем в рамках реализации концепции «Умный город». Целью работы является выбор метода обработки изображений для точного и быстрого обнаружения пешеходов при ограничении нагрузки на вычислительные ресурсы системы. Приведены основные классы решения задачи обнаружения объектов на изображениях. Показано, что традиционные методы обработки изображений являются менее требовательными и более быстрыми по сравнению с методами на основе глубокого обучения, но существенно уступают последним по точности. Выполнен анализ одноэтапных и двухэтапных моделей глубокого обучения, применяемых для обнаружения заданных объектов. По результатам анализа выбрана модель YOLOv4-tiny. На основе данной модели реализована программа, позволившая провести экспериментальные исследования по обнаружению пешеходов на изображениях. Предложенный подход показал свою эффективность и может быть использован для разработки интеллектуальных систем поддержки водителей транспортных систем.

Ключевые слова: умный город, интеллектуальная транспортная система, интеллектуальная система поддержки водителя транспортного средства, методы обнаружения объектов на изображениях, модели глубокого обучения для обнаружения объектов на изображениях.

Введение

Текущий этап развития городской инфраструктуры связан с научно-техническими направлениями в рамках концепции «Умный город» (Smart City), целью которой является повышение эффективности всех городских служб. При этом в зависимости от организации городского планирования «Умный город» можно определить как «Цифровой город», «Город знаний», «Кибергород» или «Экогород». Данная концепция подразумевает необходимость в проведении постоянного мониторинга наиболее важных объектов городской инфраструктуры, в том числе уличной дорожно-транспортной сети, мостов, туннелей, железнодорожной сети, метро, аэропортов, инфокоммуникационных систем и т.д. для оптимального распределения ресурсов и обеспечения безопасного функционирования соответствующих компонентов социально-технологической системы [1, 2].

Одной из задач «Умного города» является организация транспортных потоков. При этом весьма остро стоит проблема обеспечения безопасности участников дорожного движения. Указанная задача усложняется тем, что согласно имеющимся прогнозам через несколько лет предполагается появление на дорогах свыше 10 миллионов беспилотных транспортных средств [3]. Это обуславливает необходимость проектирования и внедрения интеллектуальных транспортных систем.

В монографии [4] дано следующее определение: «Интеллектуальная транспортная система – это интегрированная автоматизированная система, которая, используя интеллектуальные и инновационные методы организации и управления, предоставляет участникам транспортной отрасли сервисы по координированию, планированию, информированию, повышению уровней безопасности и эффективности использования транспортных сетей». Таким образом, интеллектуальные транспортные системы базируются на интеллектуальных

технологиях, что, в свою очередь, требует применения интеллектуальных бортовых автомобильных систем (отметим, что бортовые системы беспилотных автомобилей не интеллектуальными просто не могут быть). При этом необходимо, чтобы принимаемые в процессе управления решения в таких системах были по скорости и точности, по крайней мере, не хуже решений человека-водителя. Для повышения уровня интеллекта транспортных средств используют различные технологии. При этом одной из базовых технологий является компьютерное зрение, обеспечивающее получение, хранение, обработку и анализ изображений различной физической природы для извлечения информации, на основе которой осуществляется принятие решений [5].

Как правило, при описании методов и средств компьютерного зрения в интеллектуальных транспортных системах основное внимание уделяется беспилотным автомобилям [6]. Однако, несмотря на планы по увеличению количества беспилотных транспортных средств, с практической точки зрения более актуальными представляются работы в области применения указанных технологий для пилотируемых автомобилей. В этом случае говорят об интеллектуальных системах поддержки водителей (Driver Assistance) [7]. Такие системы предназначены, прежде всего, для обеспечения безопасности участников дорожного движения и рассматриваются как элементы систем активной безопасности, таким как адаптивный круиз-контроль, система автоматического торможения, система предупреждения о выезде с полосы движения и др. Их основной функцией является предотвращение или уменьшение вероятности возникновения аварий. Кроме того, еще одна цель разработки подобных систем заключается в повышении комфорта водителей. Например, неровности дороги впереди могут быть обнаружены и компенсированы бортовой системой автомобиля автоматически. Это позволяет снизить стресс водителя при поездках на дальние расстояния или по высокоскоростным автомагистралям.

Применение интеллектуальных систем поддержки водителей позволяет в значительной степени уменьшить тяжесть дорожно-транспортных происшествий, связанных с пешеходами за счет их своевременного обнаружения перед транспортными средствами. Для этого применяются различные детекторы, в том числе базирующиеся на методах машинного и глубокого обучения для обнаружения объектов на изображениях. Ввиду постоянного развития таких методов вопросы их анализа, рассмотрению которых посвящена данная статья, является актуальной и практически значимой задачей.

Методы обнаружения объектов на изображениях

На практике применяются различные методы обнаружения объектов на изображениях [8]. При этом можно выделить общие для всех базирующихся на этих методах систем компоненты, такие как база данных моделей, детектор признаков, генератор гипотез и механизм проверки гипотез [9].

Согласно статье [10] используемые в настоящее время методы обнаружения объектов на изображениях можно разделить на два больших класса – традиционные методы и методы на основе глубокого обучения. К первому классу принадлежат, например, методы Виолы-Джонса, HOG, набора слов. Ко второму классу – методы R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD.

Метод Виолы-Джонса применяется для построения одного из самых известных детекторов объектов на изображениях – каскадном классификаторе, в котором анализируются признаки Хаара или локальные бинарные шаблоны LBP [11, 12]. Классификатор имеет двухуровневую схему, в которой уровни являются независимыми. Первый уровень обеспечивает вычисление признаков в результате обработки изображения, а второй уровень обеспечивает классификацию на основе анализа вычисленных признаков. На этапе обнаружения на изображении заданного объекта (первый уровень) используется перемещаемое по изображению окно, которое определяет области для вычисления признаков. Применяемый

на втором уровне алгоритм классификации представляет собой каскад решающих деревьев (каскад отклонения). Основная операция каскада заключается в последовательном сравнении значений признаков для участков изображения с пороговыми значениями в узлах. Те участки, которые «приняты» каждым классификатором, считаются принадлежащими искомому классу, т.е. на них изображен искомый объект. Каждый узел строится так, чтобы коэффициент обнаружения был высоким (~99.9%), т.е. было мало ложноотрицательных результатов (пропущенных объектов). При этом коэффициент отклонения допускается низким (~50%), т.е. возможно большое число ложноположительных результатов – неправильно классифицированных посторонних объектов. При получении результата, который говорит о том, что участок изображения не содержит искомый объект (не принадлежит искомому классу), на любом этапе каскада приводит к прекращению его работы. Принадлежность к классу устанавливается только, если рассматриваемый участок успешно прошел через весь каскад. Если такое событие (присутствие объекта на изображении) возникает редко, то каскад отклонения обеспечивает значительное сокращение объема вычислений, т.к. для большинства участков поиск объекта быстро завершается с отрицательным результатом. Дополнительное сокращение времени обеспечивается за счет того, что самые простые (быстро обчислимые) узлы помещаются в начало каскада. В каскаде отклонения каждый узел представляет собой набор слабых классификаторов, которые посредством усиления с помощью метода AdaBoost объединяются в один сильный классификатор.

Метод HOG базируется на вычислении и анализе гистограмм ориентированных градиентов (HOG-дескрипторов), первоначально разработанных для решения задачи детектирования пешеходов на изображениях [13]. При формировании соответствующих признаков изображение разбивается на блоки, в которых вычисляются направления градиента интенсивности. Для анализа признаков применяется классификатор на основе машины опорных векторов SVM. Дальнейшее развитие метода позволило повысить точность обнаружения за счет детектирования отдельных частей тела, что обеспечивает положительный результат даже в случае смещения тела человека или его частичного перекрытия [14]. На практике положения частей относительно центра детектируемого объекта (корневого узла) неизвестны. Это так называемые латентные (скрытые) переменные модели (от них метод получил название латентного SVM). После нахождения корневого узла и других частей формулируется гипотеза об объекте, которая принимает во внимание правдоподобие того, что найденные части будут находиться в обнаруженных местах относительно корневого узла. Данный метод хорошо подходит для распознавания деформируемых объектов, потому что в явном виде реализует идею о нескольких компонентах, связанных вместе с деформируемой конструкцией. Отметим, что в настоящее время метод HOG и его модификации применяются не только для обнаружения пешеходов на изображениях, но и других объектов.

В методе набора слов соединительная структура объекта полностью игнорируется, а идея заимствована из методики распознавания документов, в которой рассматривается только список компонентов целостного объекта [15]. Таким образом, этот метод можно обобщить и использовать не только для обнаружения определенных объектов, но и для анализа целых сцен. При использовании метода предполагается построение классификатора, обучаемого на так называемых векторах присутствия. Последние отражают наличие на изображениях наиболее значимых признаков, для получения которых используются дескрипторы особых (ключевых) точек, например, дескрипторы SIFT или SURF. При анализе изображения обучающего набора определяются осмысленные кластеры особых точек (под кластерами понимаются группы похожих дескрипторов, достаточно близких в соответствующем признаковом пространстве). Для каждого кластера затем вычисляются центры, которые часто называют визуальными словами. Вся совокупность визуальных слов называется словарем. После создания словаря для каждого изображения из обучающего набора формируется вектор присутствия, представляющий собой вектор двоичных значений, равных 1 при наличии соответствующего слова или 0 при его отсутствии в словаре. После этого

классификатор обучается на полученных векторах (в качестве классификатора обычно применяют наивный байесовский классификатор или машину опорных векторов). При детектировании заданных объектов на произвольном изображении сначала осуществляется формирование вектора присутствия, а затем его анализ с помощью обученного классификатора.

В настоящее время практически во все сферы человеческой деятельности активно внедряются интеллектуальные технологии. В значительной степени эта активность связана с достижениями в области глубоких нейронных сетей (глубокого обучения). Для решения задачи обнаружения объектов на изображениях с использованием глубокого обучения рассматривают двухэтапные и одноэтапные методы. В двухэтапных методах сначала выделяются области интереса, в которых, возможно, содержатся искомые объекты, а затем эти области уточняются с помощью применения классификаторов. В одноэтапных методах поиск области интереса осуществляется одновременно с их классификацией для локализации объектов. Двухэтапные методы, например, реализуются сетями семейства R-CNN [16], а одноэтапные – YOLO [17] и SSD [18].

В нейронной сети R-CNN сначала выполняется селективный поиск областей интереса, ограниченных прямоугольными рамками различных размеров, а далее, найденные области (порядка 2000) при помощи геометрических преобразований приводятся к требуемому размеру и подаются на вход классификатора для уточнения относительно наличия искомого объекта. Отметим, что R-CNN часто строится на базе сверточной нейронной сети CaffeNet [19]. Для реализации селективного поиска может быть использован метод Bounding-Box Regression, в основе которого лежит линейная регрессия для определения ширины, высоты, центра области интереса. Базовая модель R-CNN является относительно медленной. Более высокой скоростью обладает модель Fast R-CNN [20], отличительной особенностью которой является обработка сразу всего изображения для получения признаков. Следует отметить, что здесь применяется совместное обучение классификатора SVM, сверточной нейронной сети и Bounding-Box Regression. Еще более быстрой является модель Faster R-CNN [21], в которой применяется сверточная сеть RPN. В Faster RCNN применяются Box-Regression-Layer для прогнозирования смещения ограничивающих рамок и Box-Classification-Layer для классификации области интереса. Кроме того, здесь применяются ключевые рамки с различными положениями, размерами и формами. Улучшенным вариантом Faster RCNN является Mask R-CNN [22]. Модели на базе R-CNN являются точными, но медленными. Последнее затрудняет их использование в системах реального времени, в том числе в беспилотных транспортных системах и системах поддержки водителя. Большей скоростью обладают, например, модели YOLO – по данным, приведенным на официальном сайте разработчиков алгоритма YOLO <https://pjreddie.com/darknet/yolo/>, метод YOLOv3 в 1000 раз быстрее, чем R-CNN, и в 100 раз быстрее, чем Fast R-CNN. Но следует отметить, что значение полноты обнаружения (количество правильно определенных объектов на выборке) у YOLOv3 меньше, чем у R-CNN, т.е. R-CNN работает точнее YOLOv3.

Модель YOLO была предложена в качестве первой глубокой нейронной сети, предназначенной для обнаружения объектов в реальном времени. Основной отличительной особенностью данной модели от R-CNN является отказ от использования скользящего окна. Вместо этого изображение разбивается на непересекающиеся области, относительно которых осуществляется прогнозирование о наличии центра объекта. При этом определяется несколько вариантов положения ограничивающих рамок для одного объекта, каждый из которых характеризуется координатами центра, шириной, высотой, степенью уверенности в наличии объекта и вероятностью наличия объекта. К настоящему времени разработано множество моделей YOLO. Помимо них высокой скоростью обнаружения объектов на изображениях обладает модель SSD, в которой осуществляется анализ изображений различных масштабов для принятия решения о наличии искомого объекта. В отличие от

YOLO, в SSD не предусматривается разбиение изображения на непересекающиеся фрагменты. Вместо этого прогнозируется положение ключевых рамок, которые масштабируются на различных уровнях.

Традиционные методы обнаружения объектов на изображениях отличаются достаточно высокой скоростью и не являются слишком требовательными к вычислительным ресурсам. Однако по сравнению с ними методы на основе глубокого обучения показывают более высокую точность при приемлемой скорости работы. Также к достоинствам последних можно отнести их ключевую особенность – автоматическое определение наиболее существенных признаков объектов интереса во время обучения, тогда как при использовании традиционных методов определение необходимых признаков выполняется до этапа обучения. Для эффективной работы такие методы требуют наличие обучающих выборок значительного объема. Однако для задачи обнаружения объектов дорожной обстановки в настоящее время имеется достаточно большое количество изображений. Кроме того, в настоящее время активно развиваются методы аугментации данных – расширения исходных наборов данных за счет генерации на их основе новых данных. Сказанное обуславливает выбор методов глубокого обучения для решения задачи обнаружения пешеходов на изображениях в интеллектуальных системах поддержки водителей транспортных средств.

Как было сказано, методы обнаружения объектов на основе глубокого обучения делятся на двухэтапные и одноэтапные. Двухэтапные методы, такие как R-CNN, Fast R-CNN и Faster R-CNN, обладают высокой точностью обнаружения и локализации заданных объектов на изображениях, но заложенная в них необходимость извлекать и обрабатывать обширные массивы областей-кандидатов приводит к увеличению времени вычислений и снижению эффективности. Одноэтапные методы не предусматривают выделение областей-кандидатов – в них применяется прямая передача изображения в нейронной сети для выделения признаков с последующей регрессией по ограничивающим рамкам, что обеспечивает высокую скорость работы при незначительных потерях в точности. По соотношению между скоростью и точностью одноэтапные методы лучше подходят для систем реального времени, к которым относятся интеллектуальные системы поддержки водителей транспортных средств.

Выбор метода глубокого обучения для обнаружения пешеходов на изображениях

Среди одноэтапных методов глубокого обучения для обнаружения объектов на изображениях наиболее активно развивающимися являются методы YOLO, которые с момента появления первой версии YOLOv1 стали активно применяться в системах обработки изображения дорожной обстановки реального времени [23].

К настоящему времени разработаны модели YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, Scaled-YOLOv4, YOLOR, YOLOX, YOLOv6, YOLOv7, YOLOv8, PP-YOLO, PP-YOLOv2, PP-YOLOE, DAMO-YOLO, YOLO-NAS. Кроме того, разработаны различные модификации перечисленных моделей. В обзоре, выполненном авторами работы [24], приведены данные относительно моделей YOLO, приведенные в таблице 1 (при этом необходимо учитывать, что показатели для YOLO и YOLOv2 были получены с использованием набора данных VOC 2007, а для остальных моделей – с использованием COCO 2017). Авторы данной работы указывают, что в оригинальной модели YOLO основное внимание уделялось достижению высокой скорости обнаружения объектов. В ней использовалась единая сверточная нейронная сеть для прогнозирования местоположения объектов и их классов по входному изображению, обеспечивая обработку в режиме реального времени. Однако такой акцент на скорости привел к снижению точности, в основном при работе с небольшими объектами или объектами с перекрывающимися ограничивающими рамками. В последующих версиях YOLO были внесены уточнения и усовершенствования для устранения этих ограничений при сохранении возможностей применения в режиме реального времени.

Например, в YOLOv2 (YOLO9000) были введены блоки привязки и сквозные слои для улучшения локализации объектов, что привело к повышению точности. Кроме того, YOLOv3 повысил производительность модели за счет использования многомасштабной архитектуры извлечения признаков, позволяющей лучше обнаруживать объекты в различных масштабах. Такие модели, как YOLOv4 и YOLOv5, привнесли инновации, такие как новые базовые сети, улучшенные методы увеличения объема данных и оптимизированные стратегии обучения. Эти разработки привели к значительному повышению точности без существенного влияния на производительность моделей в режиме реального времени. Начиная с YOLOv5, все официальные модели YOLO имеют оптимальное соотношение между скоростью и точностью, предлагая различные масштабы моделей в соответствии с конкретными приложениями и требованиями к оборудованию. Например, эти версии часто предлагают облегченные модели, которые в обмен на точность снижают вычислительную сложность и увеличивают скорость работы.

Таблица 1. Характеристики моделей YOLO

Модель	Год разработки	Использование блоков привязки	Фреймворк разработки	Базовая сеть	Средняя точность, %
YOLO	2015	Нет	Darknet	Darknet24	63,4 на наборе VOC 2007
YOLOv2	2016	Нет	Darknet	Darknet24	63,4 на наборе VOC 2007
YOLOv3	2018	Да	Darknet	Darknet53	36,2 на наборе COCO 2017
YOLOv4	2020	Да	Darknet	CSPDarknet53	43,5 на наборе COCO 2017
YOLOv5	2020	Да	Pytorch	YOLOv5CSP Darknet	55,8 на наборе COCO 2017
PP-YOLO	2020	Да	PaddlePaddle	ResNet50-vd	45,9 на наборе COCO 2017
Scaled-YOLOv4	2021	Да	Pytorch	CSPDarknet	56,0 на наборе COCO 2017
PP-YOLOv2	2021	Да	PaddlePaddle	ResNet101-vd	50,3 на наборе COCO 2017
YOLOR	2021	Да	Pytorch	CSPDarknet	55,4 на наборе COCO 2017
YOLOX	2021	Нет	Pytorch	YOLOX-CSPDarknet	51,2 на наборе COCO 2017
PP-YOLOE	2022	Нет	PaddlePaddle	CSPRepResNet	54,7 на наборе COCO 2017
YOLOv6	2022	Нет	Pytorch	EfficientRep	52,5 на наборе COCO 2017
YOLOv7	2022	Нет	Pytorch	YOLOv7Backbone	56,8 на наборе COCO 2017
DAMO-YOLO	2022	Нет	Pytorch	MAE-NAS	50,0 на наборе COCO 2017
Модель	Год разработки	Использование блоков привязки	Фреймворк разработки	Базовая сеть	Средняя точность, %

YOLOv8	2023	Нет	Pytorch	YOLOv8CSP Darknet	53,9 на наборе COCO 2017
YOLO- NAS	2023	Нет	Pytorch	Nas	52,2 на наборе COCO 2017

Несмотря на изложенное выше, однозначных рекомендаций, какие именно модели YOLO следует применять в конкретных системах, функционирующих в конкретных условиях, не существует. Например, в таблице 1 показано, что с использованием набора данных COCO 2017 наилучшим показателем по точности обладает модель YOLOv7. Однако применительно к обнаружению пешеходов и транспортных средств в работах [23] и [25] рекомендуется использовать собственные улучшенные модификации модели YOLOv5, а в работе [26] – YOLOR-CSP.

В статье [27] авторы предлагают метод обнаружения транспортных средств и пешеходов, основанный на YOLOv4-tiny. Приведенные ими результаты экспериментов показывают, что улучшенная модель YOLOv4-tiny обладает высокой точностью и может быть использована в интеллектуальных системах поддержки водителей автомобилей. В работе [28] описана новая аппаратная реализация YOLOv4-tiny на ПЛИС для применения в беспилотных автомобилях. Данная реализация обеспечивает сокращение вычислительных ресурсов, высокую скорость и сравнимую с другими моделями YOLO точность.

Таким образом, в качестве базовой модели для обнаружения пешеходов на изображениях в интеллектуальных системах поддержки водителей транспортных средств, обеспечивающих низкие требования к вычислительным ресурсам, высокую скорость и хорошую точность, выбрана модель YOLOv4-tiny.

Отметим также, что выбранная модель поддерживается такими крупными производителями программно-аппаратного обеспечения как Intel и NVIDIA.

Экспериментальные исследования

Для проведения экспериментальных исследований была разработана демонстрационная программа, функционирующая на персональном компьютере Acer Nitro AN515-44 под управлением 64-разрядной операционной системы Microsoft Windows 2010.

Технические характеристики персонального компьютера:

- центральный процессор: AMD Ryzen 5 4600H;
- графический процессор: Radeon Graphics;
- количество ядер процессора: 6;
- частота основного процессора: 3 ГГц;
- объем оперативной памяти: 8 Гб;
- объем жесткого диска: 256 Гб.

Демонстрационная программа реализована на языке программирования Python с использованием библиотек OpenCV и Tkinter в среде разработки Microsoft Visual Studio 2019.

В качестве входных изображений были использованы видеоданные, полученные из открытых источников в сети Internet. Пример обнаружения пешеходов на изображениях в программе с использованием модели YOLOv4-tiny показаны на рис. 1 и 2.

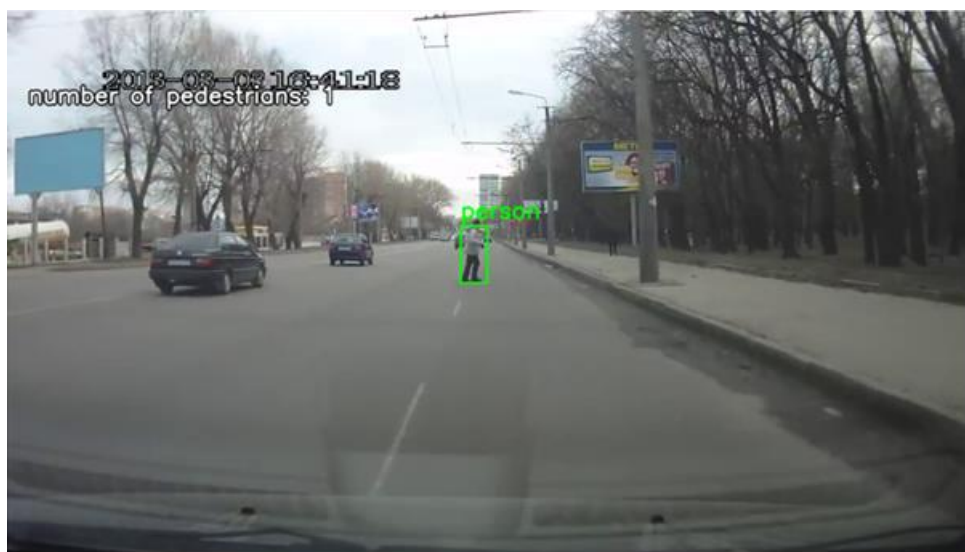


Рис.1. Пример обнаружения одного пешехода на изображении



Рис.2. Пример обнаружения двух пешеходов на изображении

При исследовании эффективности работы моделей YOLO важным настроечным параметром является параметр точности, представляющий собой пороговое значение IoU – отношение площади пересечения к площади объединения областей изображения внутри предсказанной и истинной рамок, ограничивающих искомый объект. Таким образом, параметр точности определяет некоторым образом степень уверенности в том, что объект найден, хотя возможно и не весь попал в предсказанную рамку. Это тем более важно, что объекты на реальных изображениях могут несколько отличаться от объектов на обучающих изображениях. Выбор оптимального значения данного параметра позволяет в определенный момент остановить поиск объекта, сократив тем самым время работы. На рис. 3 и 4 приведены диаграммы, отражающие влияние параметра точности на результаты обнаружения объектов на изображениях. При этом значение параметра изменялось от 0 до 1 с шагом 0,1.

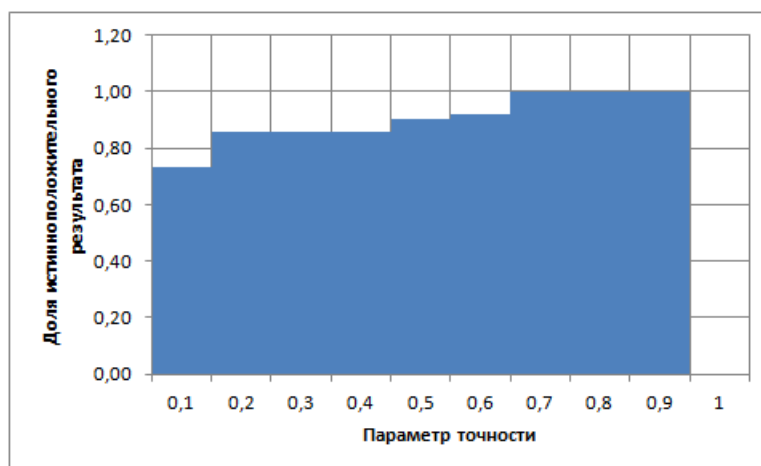


Рис. 3. Влияние параметра точности на истинноположительный результат

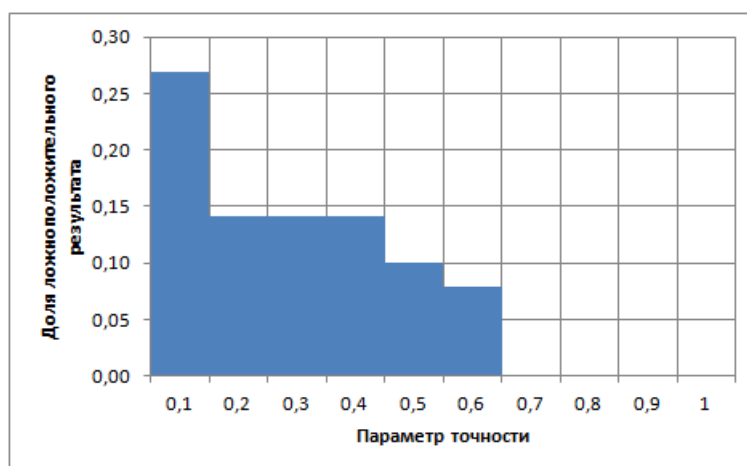


Рис. 4. Влияние параметра точности на ложноположительный результат

При проведении экспериментов в качестве входных изображений были использованы кадры цветных видеозаписей автомобильных регистраторов, полученные из открытых источников в сети Internet. При этом качество видеозаписей и количество пешеходов на их кадрах значительно различается. Анализ показал, что достаточно устойчивые результаты получаются при значении параметра точности, равном 0,7. В этом случае практически все пешеходы на изображении обнаруживаются при отсутствии ошибок, связанных со срабатыванием на других объектах.

При получении кадров видеозаписей программа предварительно преобразует их к размеру 1920×1080 пикселей. При таком разрешении время обработки одного кадра в используемой вычислительной системе составляет 0,15 секунд.

Таким образом, результаты проведенных экспериментов позволяют сделать выводы о высокой эффективности обнаружения пешеходов на изображениях с использованием модели глубокого обучения YOLOv4-tiny, что дает основание рекомендовать данный подход для использования в интеллектуальных системах поддержки водителей транспортных средств. Кроме того, следует отметить, что эксперименты проводились на базе относительно недорогой вычислительной системы, что дает основание говорить о низкой стоимости разработки, особенно в плане массового производства.

Заключение

Анализ источников и экспериментальные исследования показывают высокую эффективность по точности и времени обнаружения пешеходов на изображениях в интеллектуальных системах поддержки водителей транспортных средств при использовании метода на основе модели глубокого обучения YOLOv4-tiny. Реализация и внедрение подобных систем позволяет снизить количество дорожно-транспортных происшествий с участием пешеходов, особенно со смертельным исходом.

Использование модели YOLOv4-tiny обеспечивает снижение требований к вычислительным ресурсам при хорошем соотношении между точностью и временем. Это позволяет, в свою очередь, снизить стоимость разработки рассматриваемых систем, что весьма существенно для их массового производства.

Модель YOLOv4-tiny может быть также обучена обнаружению множества объектов, например, легковых автомобилей, грузовых автомобилей, автобусов, мотоциклов, а также заданных препятствий. Но для этого необходимо обеспечить балансировку классов в обучающей выборке, что подразумевает использование методов аугментации данных.

Список литературы

1. Намиот Д.Е. Умные города 2016 / Д.Е. Намиот // International Journal of Open Information Technologies. – 2016. - Т.4. - №1. - С. 1-3.
2. Каточков В.М. Современные тенденции развития Smart City в мире / В.М. Каточков, Г.В. Савин, Е.В. Топоркова // Вестник Удмуртского университета. Серия Экономика и право. – 2020. - 30 (3). - С. 340-345.
3. Климов А.А. Архитектура автономных (беспилотных) автомобилей и инфраструктура для их эксплуатации / А.А. Климов, О.Н. Покусаев, В.П. Куприяновский, Д.Е. Намиот // Современные информационные технологии и ИТ-образование. – 2018. - 14 (3). – С. 727-736.
4. Душкин Р.В. Интеллектуальные транспортные системы / Р.В. Душкин // М.: ДМК Пресс. - 2020. – 282 с.
5. Компьютерное зрение. Современные методы и перспективы развития. Под ред. Р. Дэвиса, М. Терка. Пер с англ. В.С. Яценкова. – М.: ДМК Пресс 2022. – 690 с.
6. Лю Ш. Разработка беспилотных транспортных средств / Лю Ш., Ли Л., Тан Ц., Ву Ш., Годье Ж.Л.; науч. ред. В. С. Яценков. Пер. с англ. П. М. Бомбаковой. – М.: ДМК Пресс 2022. – 246 с.
7. Rezaei M. Computer Vision for Driver Assistance: Simultaneous Traffic and Driver Monitoring / M. Rezaei, R. Klette // Springer, 2017. – 240 p.
8. Андриянов Н.А. Обнаружение объектов на изображении: от критериев Байеса и Неймана–Пирсона к детекторам на базе нейронных сетей EfficientDet / Н.А. Андриянов, В.Е. Дементьев, А.Г. Ташлинский // Компьютерная оптика. – 2022. - 46 (1). – С. 139-159.
9. Sharma K. A review and an approach for object detection in images / K. Sharma, T. Nileshsingh // International Journal of Computational Vision and Robotics. – 2017. – 7. – P.196-237.
10. Kaur J. Tools, techniques, datasets and application areas for object detection in an image: a review / J. Kaur, W. Singh // Multimedia Tools and Applications. – 2022. -81. – P. 38297-38351.
11. Viola P. Robust real-time face detection / P. Viola, M.J. Jones // International Journal of Computer Vision. – 2004. - 57(2). – P.137-154.
12. Suri P.K. Robust Face Detection Using Circular Multi Block Local Binary Pattern and Integral Haar Features / P.K. Suri, A. Verma // International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence. – 2011. – P. 67-71.

13. Dalal N. Histograms of oriented gradients for human detection / N. Dalal, B. Triggs // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). – 2005. – 1. – P. 886-893.
14. Felzenszwalb P. A discriminatively trained, multiscale, deformable part model // P. Felzenszwalb, D. McAllester, D. Ramanan // 2008 IEEE Conference on computer vision and pattern recognition. – 2008. – P. 1-8.
15. Кэлер А. Изучаем OpenCV 3 / А. Кэлер, Г. Брэдски; пер. с англ. А.А. Слинкина. – М.: ДМК Пресс. - 2017. – 826 с.
16. Girshick R. Rich feature hierarchies for accurate object detection and semantic segmentation. R. Girshick, J. Donahue, T. Darrell, J. Malik // 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2014. – P. 580-587.
17. Redmon J. You only look once: Unified, real-time object detection // J. Redmon, S. Divvala, R. Girshick, A. Farhadi // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 779-788.
18. Liu W. SSD: Single Shot MultiBox Detector / Liu W., Anguelov D., Erhan D. et al. ; eds. Leibe, B., Matas, J., Sebe, N., Welling, M. - Computer Vision – ECCV 2016. - Lecture Notes in Computer Science. – 2016. – 9905. – P. 21-37.
19. Jia Y. Caffe: Convolutional Architecture for Fast Feature Embedding / Y. Jia, E. Shelhamer, J. Donahue et al. // Proceedings of the 22nd ACM international conference on Multimedia. – 2014. - P. 675-678.
20. Girshick R. Fast R-CNN / R. Girshick // 2015 IEEE International Conference on Computer Vision (ICCV). – 2015. - P.1440-1448.
21. Ren S. Faster R-CNN: Towards real-time object detection with region proposal networks / S. Ren, K. He, R. Girshick, J. Sun // Advances in Neural Information Processing Systems (NIPS) 2015. – P. 91-99.
22. He K. Mask R-CNN / K. He, G. Gkioxari, P. Dollar, R. Girshick // 2017 IEEE International Conference on Computer Vision (ICCV). – 2017. – P. 2961-2969.
23. Li M. A Pedestrian Detection Network Model Based on Improved YOLOv5 / Li M.-L., Sun G.-B., Yu J.-X. // Entropy. – 2023. – 25. – P. 381. <https://doi.org/10.3390/e25020381>
24. Terven J., Cordova-Esparza D. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. arXiv 2023, 2304.00501.
25. Sun J. Vehicle And Pedestrian Detection Algorithm Based on Improved YOLOv5 / J. Sun, Z. Wang // IAENG International Journal of Computer Science. – 2023. - 50 (4). – P. 1401-1409.
26. Azevedo P. Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices / P. Azevedo, V. Santos // Robotics and Autonomous. – 2024. – 171. - 104558.
27. Li H.F. Target detection based on improved Yolov4-Tiny algorithm / H.F. Li, H.C. Yang, X.Y. Liang, H.H. Feng // Proc. SPIE. - 2023, 12509, 125092K.
28. Valadanzoj Z. High-speed YOLOv4-tiny hardware accelerator for self-driving automotive / Z. Valadanzoj, H. Daryanavard, A. Harifi // The Journal of Supercomputing. – 2023. – P. 1-26.

PEDESTRIAN DETECTION IN IMAGES BASED ON DEEP LEARNING MODEL

M.M. Lyasheva, S.A. Lyasheva, M.P. Shleymovich

Kazan National Research Technical University named after A. N. Tupolev-KAI
10, st. Karl Marx, Kazan, 420111, Russian Federation

Annotation. The paper considers the problem of detecting pedestrians in images in intelligent vehicle driver support systems. The solution of this problem is important in the development and implementation of intelligent transport systems within the framework of the implementation of the "Smart City" concept. The aim of the work is to choose an image processing method for accurate and fast detection of pedestrians with limited computing resources of the system. The main classes of solving the problem of detecting objects in images are given. It is shown that traditional image processing methods are less demanding and faster than methods based on deep learning, but significantly inferior to the latter in accuracy. The analysis of one-stage and two-stage deep learning models used to detect specified objects is performed. Based on the results of the analysis, the YOLOv4-tiny model was selected. Based on this model, a program was implemented that allowed experimental studies to be carried out to detect pedestrians in images. The proposed approach has shown its effectiveness and can be used to develop intelligent vehicle driver support systems.

Keywords: smart city, intelligent transport system, intelligent vehicle driver support system, object detection methods in images, deep learning models for object detection in images.

Статья представлена в редакцию 11 декабря 2023 г.