Нейросетевая система обнаружения знаков дорожного движения

А. В. Девяткин, Д. М. Филатов
Санкт–Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)
avdevyatkin@etu.ru, dmfilatov@etu.ru

Аннотация. **Сверточные нейронные сети показывают сегодня большие результаты в различных задачах, связанных с обработкой изображений. Такая область, как компьютерное зрение, является подходящим местом для использования нейронных сетей для получения полезной информации из изображений. В то же время, разработка автономного транспорта требует систем обработки изображений для обнаружения дорожных знаков. В работе представляется сверточная нейросетевая система для обнаружения дорожных знаков, которая представляет собой архитектуру YOLOv3, а также она сравнивается с версиями YOLOv2 и TinyYOLOv3 для оценки возможностей модифицирования для применения с учетом различных ограничений. Система обеспечивает обнаружение множественных объектов с представлением процента уверенности и описывающего прямоугольника. Для обучения используется компьютер с GPU и многопоточное расширением базы обучения путем аугментации данных.**

Ключевые слова: сети глубокого обучения; свёрточные нейронные сети; детектирование знаков дорожного движения

I. Введение

Системы компьютерного зрения в настоящее время применяются для решения широкого спектра задач. Современный уровень развития вычислительной техники и применение методов обработки изображений с помощью нейронных сетей позволяют создавать системы технического зрения, способные решать задачи анализа изображений не хуже чем человек, а в некоторых случаях данные системы могут оказаться эффективнее человека. Примерами использования технического зрения могут служить: система коррекции положения при установке маленьких SMD чипов на печатные платы, которая позволяет увеличить точность позиционирования чипов на платах и улучшить показатели качества производства [1]; система определения препятствий, встроенная в умное кресло для инвалидов [2]; система определения человеческих тел под водой, которая находясь в специальных условиях позволяет помочь поисковой команде отыскать тонущего человека [3]; система управления жестами с использованием датчика Kinect, которая представляет функционал доски для рисования и отображения [4]; система определения эмоционального состояния индивида по лицевым признакам [5]. Представленные задачи являются лишь малой долей тех возможностей, которые открывает область компьютерного зрения для создания систем автоматизации и управления.

Не исключением является также и область транспортных средств, в которой компьютерное зрение проявило себя в системах активной и пассивной помощи водителю. На данный момент полноценный автопилот еще не является обыденностью, и для того, чтобы автономные транспортные средства курсировали по дорогам городов, должно пройти время. Основными направлениями развития систем активной помощи водителю (ADAS) являются, с одной стороны, расширение функционала и качества работы систем с целью повышения безопасности дорожного движения, а с другой стороны, оптимизация алгоритмов и аппаратной части с целью удешевления конечного продукта, что позволит приблизить момент массового выпуска продукта на рынок.. Так, например, исследование оптимизации системы ADAS позволило как усовершенствовать цикл разработки программного обеспечения для указанной системы, так и увеличить производительность алгоритма, что, безусловно, положительно сказывается на работе всей системы [6]. Примерами исследований, направленных на расширение функционала и улучшение качества, могут служить системы распознавания дорожной разметки [7] и системы дополненной реальности, позволяющие отображать информацию на видимой для водителя области [8].

Трудно не отметить растущие интерес и успех в области глубокого обучения, которая является подобластью более широкой науки под названием “машинное обучение”. В основе подходов обучения с учителем лежит принцип самостоятельной настройки системы, ее параметров, на основе информации, поступающей в виде примеров входа и результирующего выхода как реакции на данный вход. Использование такого подхода дало толчок в развитии методов обработки изображения с использованием нейронных сетей. Сложная структура сетей позволяет выполнять операции нелинейного преобразования, что в результате позволяет решать одну из задач систем ADAS – распознавание дорожных знаков. Наиболее распространенным подходом является решение задачи распознавания знаков с использованием свёрточных нейронных сетей [9]. Не исключаются также и разнообразные модификации с использованием классических методов HOG (Histogram of Oriented Gradients) [10] и нейронных сетей небольшого

размера для уменьшения нагрузки на вычислительные ресурсы [11].

В работе представлена разработка системы детектирования дорожных знаков с использованием подхода глубокого обучения. Наибольший интерес по характеристикам представляет скорость обработки изображений, а также качество оценки, в которое входит как точность локализации знака, так и корректная оценка размеров знаков. В качестве базы данных был взят открытый набор под названием German Traffic Sign Detection Benchmark с 900 изображениями кадров дорожной обстановки.

II. Описание модели

Из современных моделей для решения задачи детектирования объектов в режиме реального времени можно выделить двух основных фаворитов: YOLO (You Only Look Once), на данный момент существует уже третья версия [12], и SSD (Single Shot Detector) [13]. В качестве меры оценки для моделей, задачей которых является детектирование объектов, используются датасеты ImageNet, COCO и VOC 2007 и 2012 годов. Сравнение производительности двух выделенных моделей показывает, что YOLO при большей скорости обработки (46 vs 244 FPS) имеет меньшую относительно SSD точность детектирования (41.2 vs 23.7 mAP) [14]. В свою очередь, начиная со второй модификации, YOLO превосходит как в скорости, так и в точности SSD при определенных модификациях сети, что является решающим фактором для выбора базового метода решения. Таким образом, за основу решения была выбрана модель YOLOv2, с которой сравниваются модели YOLOv3 и TinyYOLOv3 при решении задачи детектирования дорожных знаков. Для выравнивания условий сравнения моделей процесс обучения использовался в соответствии с описанным в последней версии сети [12].

В основе работы моделей YOLO лежит принцип наложения сетки на изображение (рис. 1), в которой каждая ячейка имеет ряд параметров, являющихся выходными данными сети. Данных параметров достаточно для определения описывающего прямоугольника объекта, класса объекта и параметра “уверенности” сети в принятом решении.

Размер выходного пространства определяется следующими параметрами:

- Количество ячеек по ширине (W) и высоте (H)
- Количество базовых прямоугольников (A)
- Количество предсказываемых классов объектов (C)

Под базовыми (опорными) прямоугольниками понимаются ряд определенных соотношений ширины и высоты (рис. 2), которые корректируются сетью для получения результирующего описывающего прямоугольника объекта. В данной работе используется три опорных прямоугольника на каждый выход сети.

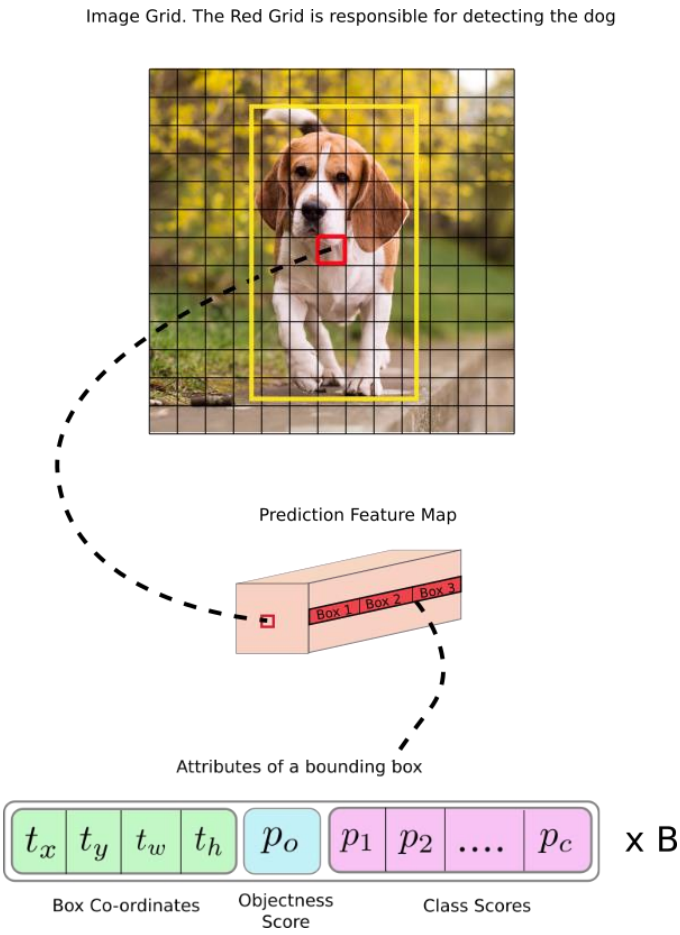

Рис. 1. Представление выходной сетки архитектуры YOLO (https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/)

Количество предсказываемых классов объектов подразумевает ширину набора классов предсказаний (кот, собака, самолет, …). Для данной работы происходит определение единственного класса объектов на кадре – знаки.

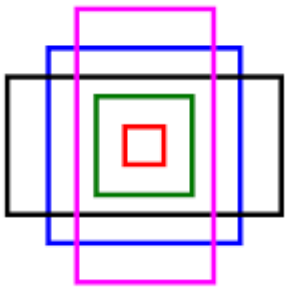

Рис. 2. Пример базовых прямоугольников

Формула для вычисления выходных параметров выглядит следующим образом: W * H * A * (4 + 1 + C). В данной реализации использовалась сетка 13 на 13 ячеек, три базовых прямоугольника, единственный класс предсказания. Таким образом, количество выходных параметров равно 3042.

Определение количества параметров важно для постобработки, так как для каждый кадр предсказывает W * H * A = 507 описывающих прямоугольников с различной степенью уверенности. Для дальнейшей постобработки алгоритмом NMS (Non Maximum Supression) количество предсказанных описывающих прямоугольников является критичным в виду вычисления пересечения с каждым из них для определения, какие из них требуется отбросить.

Алгоритм NMS представляет собой алгоритм исключения повторений предсказаний на основе определенного признака, объединяющего несколько предсказаний. Сеть YOLO во всех версиях генерирует большое количество предсказаний, которые объединяются в группы на основе признака пересечения IOU. Далее из группы остается единственное предсказание, которое имеет наибольший показатель уверенности, а остальные предсказания удаляются.

Для корректной работы нейронной сети важным шагом является процесс предобработки изображения перед передачей данных на вход. Один из важных факторов — это использование масштабирования изображения с сохранением пропорций ширины к высоте. Так как растяжение изображения на размер входа нейронной сети может искажать форму знака в зависимости от соотношения размеров сырых данных, такой прием позволяет сохранить форму знака и уменьшить влияние исходных данных на корректность работы сети.

В процессе предобработки производится нормирование данных, представляющее собой преобразование диапазона цветовых каналов от [0; 255] к [-1.0; 1.0].

Остаточные области, которые появляются из-за сохранения соотношения размеров, заполняются нулями, что в совокупности с нормированием представляет центральное состояние входного диапазона сети.

Расширение базы данных обучения было реализовано с использованием подхода аугментации данных. В качестве основных методов аугментации были использованы приемы, описанные в табл. I. Примеры полученных изображений представлены на рис. 3.

ТАБЛИЦА I МЕТОДИКИ АУГМЕНТАЦИИ ДАННЫХ

Метод аугментации	Параметры метода
Масштабирование	[25; 200] %
Сдвиг	[0; 30] %
Горизонтальное отражение	с вероятностью 50 %
HSV размытие	[100; 150] %

Разработка модели велась с использованием фреймворка для создания нейронных сетей Keras на основе backend Tensorflow на языке Python. Обучение модели проводилось с использованием аппаратного ускорения на GPU Nvidia GTX 1080. Оценки скорости работы также производились на указанном GPU.

III. ОПРЕДЕЛЕНИЕ ЗНАКА

Для тестирования и конечной валидации датасет из 900 изображений был разделен соотношением 1 к 8, в результате было получено 100 тестовых и 800 тренировочных изображений. Алгоритм NMS разделяет описывающие прямоугольники по признаку 0.5 IOU и производит постобработку только выходных данных с уверенностью более 0.5. В качестве метрики для оценки была использована метрика соревнований VOC под названием mean average precision (mAP). При использовании единственного класса данная метрика преобразована к метрике средней по выборке точности (AP).

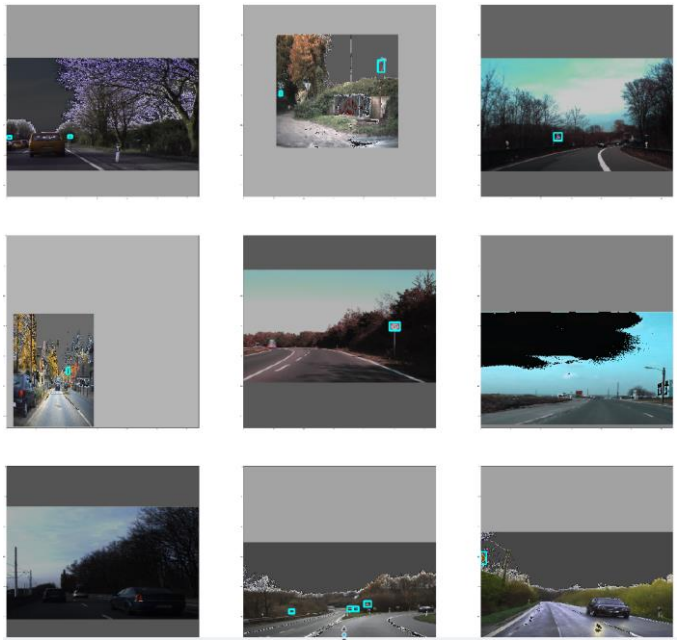

Рис. 3. Примеры аугментации данных

Обучение сетей в течении 1000 эпох с увеличением пула данных в четыре раза привело к следующим результатам, приведенным в таблице II.

ТАБЛИЦА II РЕЗУЛЬТИРУЮЩИЕ ПОКАЗАТЕЛИ СЕТЕЙ

Название сети	Средняя точность (AP), %	Скорость работы, FPS
YOLOv2 (Darknet19)	68.5	26.45
YOLOv3 (Darknet53)	84.0	16.5
TinyYOLOv3	66.0	37.33

Базовая сеть YOLOv2 имеет 19 слоев, поэтому feature extraction блок сети назван Darknet19. Сама по себе сеть имеет хорошую для реального времени производительность и точность детектирования.

Следующие версии сети YOLO показывают более высокие характеристики как по точности, так и по скорости. Полноформатная YOLOv3 с feature extraction блоком под названием Darknet53 по аналогии с предшественником имеет 53 слоя. Выход сети имеет три сетки с размерами 13x13, 26x26 и 52x52. Увеличение количества слоев, а также большее количество ячеек сеток позволяет увеличить точность определения мелких объектов на изображении. В свою очередь архитектура

TinyYOLOv3 имеет сокращенную версию блока получения классификационных признаков без названия и имеет две сетки 13x13 и 26x26, что позволяет держаться на уровне полноформатного предшественника, но при этом сильно повысить скорость обработки. Результаты работы сети представлены на рис. 4.

Рис. 4. Результат работы сети архитектуры TinyYOLOv3

IV. Заключение

В работе были рассмотрены особенности разработки детектора знаков на основе семейства нейронных сетей YOLO. Представленные результаты демонстрируют рост производительности и точности детектирования. При этом, в зависимости от требований, может быть выбрана одна из рассмотренных версий сетей, которые показывают выделяющиеся характеристики относительно предшествующей версии. Детектор показывает близкий к ста процентам результат на изображениях знаков с четкой видимостью. Дальнейшая модернизация базы обучения, а также методов предобработки и постобработки, предполагает улучшение точности детектирования. Использованные подходы будут встроены в полноформатные автомобилеподобные роботы для решения задач автономного движения и помощи оператору.

Список литературы

[1] Fernandez Elian, Farkhad Ihsan Hariadi, Muhammad Iqbal Arsyad, “Implementation of computer vision algorithms for position correction of chip-mounter machine”, IEEE, 2017 International Symposium on Electronics and Smart Devices (ISESD), 11 January 2018, DOI: 10.1109/ISESD.2017.8253311

[2] Fitri Utaminingrum, M Ali Fauzi, Randy Cahya Wihandika, Sigit Adinugroho, Tri Astoto Kurniawan, Dahnial Syauqy, Yuita Arum Sari, Putra Pandu Adikara, “Development of computer vision based obstacle detection and human tracking on smart wheelchair for disabled patient”, IEEE, 2017 5th International Symposium on Computational and Business Intelligence (ISCBI), 02 October 2017, DOI: 10.1109/ISCBI.2017.8053533

[3] Fatma Günseli YaŞar, Hüseyin KusetoĞullari, “Underwater human body detection using computer vision algorithms”, IEEE, 2018 26th Signal Processing and Communications Applications Conference (SIU), 09 July 2018, DOI: 10.1109/SIU.2018.8404305

[4] Ahtasham Ahsan, Shaheryar Najam, Jameel Ahmed, Zohaib Najam, “Interactive white board using gestures with KINECT”, IEEE, 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 24 November 2016, DOI: 10.1109/ICEEOT.2016.7754805

[5] Loredana Stanciu, Florentina Blidariu, “Emotional states recognition by interpreting facial features”, IEEE, 2017 E-Health and Bioengineering Conference (EHB), 31 July 2017, DOI: 10.1109/EHB.2017.7995414

[6] Marcos Nieto, Gorka Vélez, Oihana Otaegui, Seán Gaines, Geoffroy Van Cutsem, “Optimising computer vision based ADAS: vehicle detection case study”, IEEE, IET Intelligent Transport Systems (Volume: 10, Issue: 3, 4 2016), 28 March 2016, DOI: 10.1049/iet-its.2014.0303

[7] Sanket J. Mankar, Manoj Demde, Prashant Sharma, “Design of computer vision intelligent system for lane detection”, IEEE, 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 04 May 2017, DOI: 10.1109/GET.2016.7916843

[8] Devi Chilukuri, Sun Yi, Younho Seong, “Development of Mobile Application for VRUs Using Computer Vision”, IEEE, SoutheastCon 2018, 04 October 2018, DOI: 10.1109/SECON.2018.8479138

[9] H. Ohara, I. Nishikawa, S. Miki, N. Yabuki, “Detection and recognition of road signs using simple layered neural networks”, IEEE, Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02., 13 May 2003, DOI: 10.1109/ICONIP.2002.1198133

[10] R. Rajesh, K. Rajeev, K. Suchithra, V.P. Lekhesh, V. Gopakumar, N.K. Ragesh, “Coherence vector of Oriented Gradients for traffic sign recognition using Neural Networks”, IEEE, The 2011 International Joint Conference on Neural Networks, 03 October 2011, DOI: 10.1109/IJCNN.2011.6033318

[11] Mian Mian Lau, King Hann Lim, Alpha Agape Gopalai, “Malaysia traffic sign recognition with convolutional neural network”, IEEE, 2015 IEEE International Conference on Digital Signal Processing (DSP), 10 September 2015, DOI: 10.1109/ICDSP.2015.7252029

[12] Redmon, Joseph and Ali Farhadi. “YOLOv3: An Incremental Improvement.” CoRR abs/1804.02767 (2018)

[13] Liu, Wei et al. “SSD: Single Shot MultiBox Detector.” Lecture Notes in Computer Science (2016): 21–37. Crossref. Web.

[14] YOLO: Real-Time Object Detection, URL: https://pjreddie.com/darknet/yolo/ (Access date: 30.11.2018)