

Оптимизация модели нейронной сети U-Net-70 для сегментации железнодорожного пути



П. Е. Машченко,
канд. техн. наук,
зам. генерального
директора
ООО «ЛокоТех-Сигнал»



П. П. Ширяев,
специалист
по компьютерному
зрению
ООО «ЛокоТех-Сигнал»

В статье представлен обзор вариантов применения сегментационных нейронных сетей для беспилотного железнодорожного транспорта и главные этапы оптимизации работы модели нейронной сети на примере архитектуры U-Net-70 [1]. Проанализированы основные особенности обучения и работы этой нейронной сети.

Семантическая сегментация — это процесс получения информации из изображения. Данная задача машинного зрения заключается в разбиении изображения на области, соответствующие различным объектам. Результат сегментации — попиксельная маркировка изображения, дифференцирующая его на заранее определенные классы. Данный метод позволяет распознать сцену изображения и выделить интересные участки, необходимые для дальнейшей обработки и вычленения информации [2].

В железнодорожной сфере объектами, представляющими наибольший интерес для сегментации, являются: железнодорожные пути, рельсы, железнодорожные переезды, крестовины и остряки стрелочных переводов, поезда и другие объекты инфраструктуры. Распознавание данных элементов на изображении обеспечивает понимание того, в каком пространстве движется локомотив и что его окружает.

Трудность этой задачи в том, что железнодорожная карта Российской Федерации включает в себя множество природных

и климатических зон, а в объектив камеры могут попадать различные ландшафты и участки железнодорожного пути. Также на изображение влияют изменчивость погодных условий, время суток, время года. Затрудняет задачу наличие сложных и редких участков пути, для которых необходим дополнительный сбор данных: тоннели, мосты, депо, производственные цеха. В зависимости от типа локомотива (маневровый или магистральный) и его скорости движения накладываются свои требования, касающиеся дальности определяемого пути. Поэтому камеры системы машинного зрения могут различаться как по техническим характеристикам (разрешение, фокусное расстояние, дисторсия), так и по расположению (угол наклона, высота от земли), выдавая разную картинку.

В настоящий момент лидирующие позиции в решении этой проблемы заняли методы глубокого обучения, демонстрирующие высокое качество работы и обеспечивающие робастность при большой вариативности входных данных. Нейронные сети (НС) обрабатывают множество изображений в ходе своего обучения, тренируясь вычленять



Рис. 1. Железнодорожный путь с габаритными зонами и препятствием



Рис. 2. Предсказание траектории движения локомотива по определенному нейронной сетью железнодорожному пути

и генерализировать характерные особенности каждого предсказываемого класса. Сегментационные НС учатся выявлять, к какому классу принадлежит каждый пиксель изображения, тем самым позволяя не только идентифицировать объекты, но и определять их местоположение [3].

К объектам дорожной сцены относятся: дорога, человек, автомобиль, объекты инфраструктуры и др. [4]. При проектировании беспилотных локомотивов наибольшее внимание уделяется определению железнодорожного полотна. Принятие решения о дальнейшем движении автономным транспортом напрямую зависит от распознавания того, какие объекты или препятствия находятся на его пути. Сегментация изображения при помощи НС позволяет выделить железную дорогу для проведения дальнейшего анализа, связанного непосредственно только с ней. Таким образом, сегментационные алгоритмы стали неотъемлемой частью беспилотных систем [5].

Стоит отметить, что распознавание такого объекта, как человек или автомобиль, возможно не только при помощи сегментации. С данной задачей успешно справляются детектирующие НС. Они определяют границы объекта на изображении [6]. Однако этот алгоритм не подходит для поиска столь крупных областей изображения, как железнодорожный путь. В таком случае в границы объекта попадают и другие предсказываемые классы. Результат подобного определения становится малоинформативным.

Сегментация железнодорожных путей позволяет устанавливать габаритные зоны, внутри которых обнаруживаются препятствия, например: автомобиль на переезде; человек, переходящий пути в неположенном месте (рис. 1). При помощи сегментации элементов стрелочных переводов определяется положение перевода и строится дальнейшая траектория движения локомотива (рис. 2) [7]. Хотя сегментационные НС позволяют выполнять мультиклассовую сегментацию, в рамках статьи, в качестве примера, будет рассмотрено предсказание только одного класса — железнодорожный путь.

Параметр оптимизации нейронной сети

Задача оптимизации НС для беспилотного транспорта — поиск наилучшего сочетания точности НС и скорости работы. Точность определяется тем, насколько правильно были предсказаны пиксели на изображении, относящиеся к определяемому объекту. Как показывает практика, наиболее проблемная зона для сегментации — граница объектов. Ее соответствие истинному

Табл. 1. Изменения точности и скорости работы при разных вариантах квантизации весов нейронной сети

	Pytorch, float32	TensorRT, float32	TensorRT, float16	TensorRT, float16 + int8
mIoU, %	90,9	90,9	90,9	90,9
FPS	110	171	286	322
VRAM, Мб	2535	1777	1551	1537

контуру объекта зачастую вносит основной вклад в расчет метрики качества модели НС. Стоит отметить, что четкое определение границ объекта важно для понимания их расположения в дорожной сцене, но не критично. Например, незначительное отклонение маски железнодорожного пути от края рельс не мешает определить пространство (рис. 3), где появление препятствия способно привести к аварийной ситуации. Необходимая точность сегментационной НС устанавливается так, чтобы эта НС в комплексе с другими элементами беспилотной системы обеспечивала требуемый уровень безопасности.

Кроме того, точность сегментации НС зависит от количества данных, на которых обучается НС, и используемых гиперпараметров. К выбираемым гиперпараметрам относятся: алгоритм градиентного спуска, функция потерь, скорость обучения, объем данных на шаг обучения, методы аугментации и мн. др. Не существует однозначного и общепризнанного решения по выбору гиперпараметров. Для любых данных задействуется экспериментальный поиск оптимальных гиперпараметров, обеспечивающих требуемую точность предсказания. Мы описываем те характеристики обучения НС, которые позволили нам добиться наилучшего качества сегментации. Выбор многих из представленных параметров основан на успешном опыте их применения, зафиксированном как в наших исследованиях, так и в опубликованных текстах других создателей систем компьютерного зрения. Использование нашей конфигурации гиперпараметров, вероятнее всего, позволит обеспечить достойное качество работы и на других данных. Однако стоит помнить, что для любой задачи наиболее эффективная комбинация параметров всегда индивидуальна.

Однозначная цель оптимизации — скорость работы НС; она определяет, какое количество кадров за единицу времени модель способна обработать. Следовательно, ускорение работы НС позволяет повысить плотность предсказаний, тем самым увеличивая объем обработанных данных и повышая робастность всей системы компьютерного зрения автономного локомотива.

Обоснование выбора архитектуры нейронной сети

Выбор архитектуры сегментационной НС зависит от условий ее конечного применения. Глубина архитектуры, выполняемые математические операции, количество слоев и их типы влияют на точность работы НС. Также ограничением является производительность видеокарты или устройства, на котором выполняются все вычисления для НС.

Чтобы системы машинного зрения локомотива функционировали успешно, требуется быстрая обработка кадров, полученных с камер, нейронной сетью. Время реакции машиниста на аварийную ситуацию составляет от 0,5 с [8]. Но зачастую человеку достаточно единственного изображения, чтобы однозначно охарактеризовать ситуацию на дороге и принять решение. В случае предсказания НС возможны ложные срабатывания. Поэтому при проектировании беспилотных систем важно, чтобы финальное предсказание основывалось на результате обработки нескольких последовательных кадров, тем самым отфильтровывая ложные срабатывания. Подобная практика повышает надежность реакции системы на возникающие препятствия. Поэтому для автономного транспорта стараются применять такие НС, чтобы их скорость обработки информации была соотносима с частотой съемки камеры (например, 20–30 кадров в секунду для камер с интерфейсом GigE).

Следовательно, при выборе архитектуры нужно учитывать все эти факторы и находить баланс между точностью работы НС, скоростью работы НС и количеством ресурсов, требуемых для работы НС.

В настоящий момент существует большое количество архитектур сегментационных НС, обладающих своими преимуществами по каждому из ранее перечисленных пунктов [4; 9]. В нашем случае выбор архитектуры был основан на результатах скорости и точности работы моделей НС на тестовых данных Cityscapes [10]. Cityscapes — один из популярных датасетов, позволяющих сравнивать эффективность различных реализаций сегментационных НС. Возможно, более репрезентативным было бы использование датасета RailSem19, состоящего из аннотированных изображе-

ний с камер внутри кабины машиниста [11]. Однако пока на этом датасете не проводится сравнение результатов работы нейронных сетей [12]. Рейтинг же Cityscapes составлен для НС, решающих задачу сегментации в режиме реального времени и позволяющих сравнить более 20 различных архитектур. Также стоит отметить, что это не обязательная, но закономерная практика, когда одна и та же НС способна показывать достойный результат на разных типах данных [13]. Поэтому можно предполагать, что НС, умеющая хорошо предсказывать автомобильную дорогу, после переобучения на новых данных сможет справиться с предсказанием железнодорожного пути.

Лидирующую позицию по результатам работы на датасете Cityscapes занимает архитектура U-HarDNet-70. Эта нейронная сеть демонстрирует наивысшую точность работы по мере Жаккара при высокой скорости обработки изображения (53 кадра в секунду на изображениях с разрешением 1024×2048 пикселей на видеокарте NVIDIA GeForce GTX 1080Ti) [14].

U-HarDNet-70 — сегментационная сверточная нейронная сеть U-net-типа [15], в основании которой лежит архитектура HarDNet. HarDNet унаследовала свою архитектуру от семейства нейронных сетей DenseNet [16] и переняла ряд идей от LogDenseNet [17] и SparseNet [18]. Особенность HarDNet — применение новых оптимизационных приемов: новый вид разрежения связей между слоями (рис. 4) и новые комбинационные изменения сверточных слоев. Данные модификации позволили увеличить скорость работы НС — по сравнению с ее прародителями — без потери качества [1].

Таким образом, учитывая все вышеперечисленные преимущества U-HarDNet-70, эта архитектура — отличный выбор в качестве сегментационной НС для систем машинного зрения, работающих в реальном времени.

Обучение нейронной сети

Данные для обучения и тестирования НС были собраны самостоятельно (42 %) либо взяты из открытого датасета RailSem19 (58 %). RailSem19 содержит 8500 размеченных изображений с камер, установленных на локомотивах и трамваях в 38 странах мира. Изображения получены при различных погодных условиях и в разные сезоны. Сюда не были включены изображения низкого качества и снятые с трамвая. Наши данные были собраны системами машинного зрения, установленными на маневровых локомотивах, курсирующих на производственных предприятиях Российской Федерации. Включены все времена года, разное время суток и черно-белые фотографии с монохромных камер.

Общее количество аннотированных изображений: 10424. 1 тыс. снимков сформировала тестовую выборку. Оставшиеся изображения с масками железнодорожного пути были разбиты — в соотношении 95 к 5 % — на датасеты для обучения/валидации.

Для расширения количества данных была применена аугментация «на лету». Эта методика позволяет изменять характеристики изображения во время обучения НС, тем самым генерируя новые данные из уже имеющихся. Применение аугментаций при обучении НС позволяет повысить точность работы на десятки процентов [19], и потому упомянутый прием — один из основополагающих при оптимизации алгоритма сегментации. Все аугментации, кроме горизонтального поворота, были разбиты на группы в зависимости от идеи их преобразования. Такими группами стали: искажение изображения (cutout [20], оптические искажения и др.); смещение изображения (поворот на небольшой угол, масштабирование и пр.); цветовые искажения (изменение гаммы, яркости, цвета канала и т. д.); размытие (гауссово размытие, размытие от движения и т. п.); добавление шума (гауссов шум, понижение качества изображения и др.). Из этих групп формирова-

лась единая аугментационная композиция, где могло быть выбрано одно изменение изображения из группы с 50-процентной вероятностью.

В качестве функции потерь была выбрана комбинация бинарной перекрестной энтропии и функции потерь Дайса. Функция Дайса используется для вычисления перекрытия между той маской железнодорожного пути, что предсказана нейронной сетью, и той, что аннотирована человеком, которая принимается за истинную. В комбинации с кросс-энтропией функция потерь Дайса позволяет показывать более стабильный и точный результат обучения НС [21]. Весовые коэффициенты каждой компоненты функции потерь составили 0,5. В роли алгоритма минимизации функции потерь выступал Ranger, являющийся сочетанием алгоритмов Radam [22] и Lookahead [23]. Скорость обучения стартовала со значения 0.001. Если точность на валидационной выборке не повышалась в течение трех эпох, то скорость обучения умножалась на 0.1. Нейронная сеть обучалась на протяжении 100 эпох. Для финальной модели выбирались веса связей с той эпохи, где результат предсказания был наилучшим. Обучение модели происходило при помощи фреймворка Pytorch.

Повышение производительности нейронной сети

При внедрении модели глубокого обучения в систему беспилотного транспорта всегда стоит задача обеспечения высокой производительности НС. Снижение количества ресурсов компьютера, потребляемых моделью НС, позволяет сделать систему более энергоэффективной, а оптимизация ее работы под определенную аппаратную конфигурацию вычислителя повышает скорость предсказания сегментационных масок для изображений.

Существуют различные методы и приемы, дающие возможность повысить скорость работы НС: квантизация весов

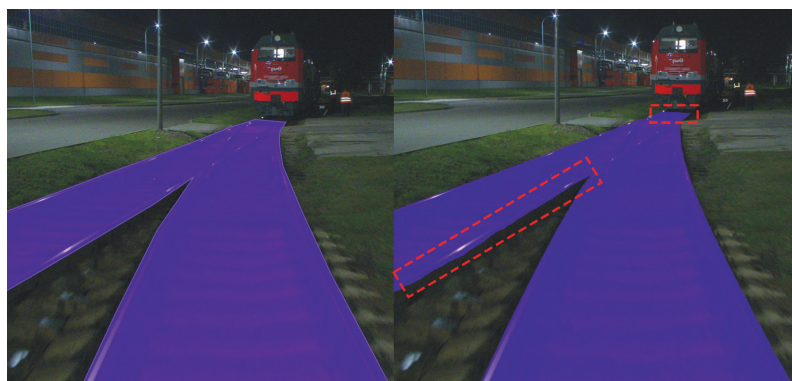


Рис. 3. Аннотированная и предсказанная сегментационная маска железнодорожного пути. Красной пунктирной линией указаны наиболее различающиеся области

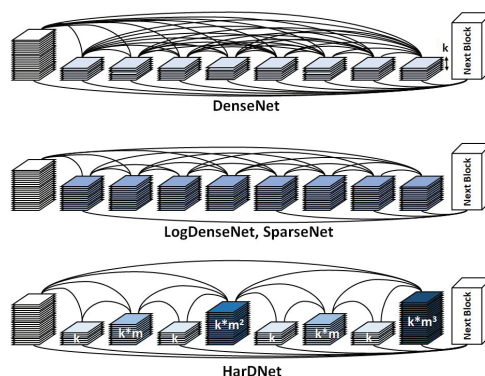


Рис. 4. Схема разрежения связей между слоями в архитектуре HarDNet по сравнению с DenseNet, LogDenseNet и SparseNet

модели, слияние и удаление слоев, подбор наиболее эффективных сверточных ядер и др. Высокопроизводительный ускоритель NVIDIA TensorRT автоматически оптимизирует обученную нейронную сеть для максимальной эффективности работы на используемой в системе видеокарте.

Также скорость работы НС зависит от размера и количества входных изображений. В рамках данной статьи эксперименты по предсказанию проводились только для одного изображения в определенный момент времени. Уменьшение размера приводит к уменьшению объема информации, которую приходится обрабатывать НС. Поэтому обрезка изображения, где нет железнодорожного пути, например часть кадра выше линии горизонта, — еще один из способов ускорить работу сегментационного алгоритма. В наших тестах участвовали изображения, разрешение которых было уменьшено с 2048×2432 до 1024×1216 пикселей.

В табл. 1 продемонстрированы изменения точности и скорости работы при разных вариантах квантизации весов НС. Точность оценивалась по среднему арифметическому значению мер Жаккара (mIoU) по всем изображениям тестовой выборки. Скорость работы НС — количество кадров в секунду (FPS), по которым способна выполнить предсказания НС, без учета формирования тензора из изображения и последующей обработки результата. Все тесты проводились на видеокарте NVIDIA GeForce GTX 2080Ti.

Квантизация — это метод для уменьшения размера нейронной сети в оперативной памяти. Суть метода заключается в том, что все множество весов сети разбивается на 256 интервалов [24]. Каждому интервалу ставится в соответствие 8-битный (int8) или 16-битный тип данных (float16) взамен 32-битного (float32). Так происходит редуцирование памяти, необходимой для работы НС (VRAM), повышается скорость выполнения операций и вырастает общая производительность.

Во время оптимизации TensorRT самостоятельно анализирует, у каких слоев НС можно уменьшить размерность для повышения скорости работы без потери качества. Согласно указаниям создателей TensorRT, связка float16 + int8 позволяет обеспечить наилучшую производительность, поскольку в этом случае есть возможность определить, какая квантизация оптимальна для конкретного слоя [25]. Однако для выполнения квантизации с int8 на вход ускорителю требуется калибровочный датасет. Полная квантизация

весов модели к 8-битному представлению не рассматривалась, так как это может приводить к потере точности предсказания [26]. Дальнейшее, более детальное профилирование работы модели можно проводить с помощью инструмента NVIDIA Nsight Systems.

В предложенном эксперименте оптимизация модели позволила получить ускорение работы НС практически в три раза по сравнению с исходной моделью — без потери точности. Также столь высокий FPS — по сравнению с результатами НС на Cityscapes — объясняется техническими характеристиками выбранной видеокарты. Она имеет большее количество ядер CUDA, нежели NVIDIA GeForce GTX 1080Ti (4352 против 3584). Но главная особенность 2080Ti — более высокая вычислительная способность (compute capability; 7.5 против 6.1), предоставляющая больше возможностей для эффективной реализации вычислений. ■

Литература:

1. Chao P. et al. Hardnet: A Low Memory Traffic Network // Proceedings of the IEEE International Conference on Computer Vision. 2019. Pp. 3552–3561.
2. Lateef F., Ruichek Y. Survey on Semantic Segmentation Using Deep Learning Techniques // Neurocomputing. 2019. Vol. 338. Pp. 321–348.
3. Kaymak Ç., Uçar A. A Brief Survey and an Application of Semantic Image Segmentation for Autonomous Driving // Handbook of Deep Learning Applications. Cham: Springer, 2019. Pp. 161–200.
4. Feng D. et al. Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges // IEEE Transactions on Intelligent Transportation Systems. 2020. Pp. 1–20.
5. Janai J. et al. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art // Foundations and Trends® in Computer Graphics and Vision. 2020. Vol. 12. No. 1–3. Pp. 1–308.
6. Zou Z. et al. Object Detection in 20 Years: A Survey. URL: arxiv.org/pdf/1905.05055.pdf (дата обращения: 10.12.2020).
7. Романчиков А. М. и др. Определение траектории движения локомотива для систем машинного зрения // Железнодорожный транспорт. 2020. № 11. С. 61–63.
8. Козубенко В. Г. Безопасное управление поездом: вопросы и ответы. М.: Маршрут, 2005.
9. Liu X., Deng Z., Yang Y. Recent Progress in Semantic Image Segmentation // Artificial Intelligence Review. 2019. Vol. 52. No. 2. Pp. 1089–1106.
10. Cordts M. et al. The Cityscapes Dataset for Semantic Urban Scene Understanding // Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition. 2016. Pp. 3213–3223.

11. Zedel O. et al. RailSem19: A Dataset for Semantic Rail Scene Understanding // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019. Pp. 1221–1229.
12. URL : <https://wilddash.cc/railsem19> (дата обращения 28.12.2020)
13. Zhuang F. et al. A Comprehensive Survey on Transfer Learning. URL: arxiv.org/pdf/1911.02685.pdf (дата обращения: 10.12.2020).
14. URL: paperswithcode.com/sota/real-time-semantic-segmentation-on-cityscapes (дата обращения: 10.12.2020).
15. Ronneberger O., Fischer P., Brox T. U-net: Convolutional Networks for Biomedical Image Segmentation // International Conference on Medical Image Computing and Computer-assisted Intervention. Cham: Springer, 2015. Pp. 234–241.
16. Huang G. et al. Densely Connected Convolutional Networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. Pp. 4700–4708.
17. Hu H. et al. Log-DenseNet: How to Sparsify a DenseNet. URL: openreview.net/pdf?id=SJLy_SxC- (дата обращения: 10.12.2020).
18. Zhu L. et al. Sparsely Aggregated Convolutional Networks // Proceedings of the European Conference on Computer Vision (ECCV). 2018. Pp. 186–201.
19. Shorten C., Khoshgoftaar T. M. A Survey on Image Data Augmentation for Deep Learning // Journal of Big Data. 2019. Vol. 6. No. 1. Pp.
20. DeVries T., Taylor G. W. Improved Regularization of Convolutional Neural Networks with Cutout. URL: arxiv.org/pdf/1708.04552.pdf (дата обращения: 10.12.2020).
21. Isensee F. et al. Automated Design of Deep Learning Methods for Biomedical Image Segmentation. URL: arxiv.org/pdf/1904.08128.pdf (дата обращения: 10.12.2020).
22. Liu L. et al. On the Variance of the Adaptive Learning Rate and Beyond. URL: openreview.net/pdf/4dbb68ed4eb409f773d227c4c157de6b31b3080b.pdf (дата обращения: 10.12.2020).
23. Zhang M. et al. Lookahead Optimizer: K Steps Forward, 1 Step Back // Advances in Neural Information Processing Systems. 2019. Pp. 9597–9608.
24. Grachev A. M., Ignatov D. I., Savchenko A. V. Compression of Recurrent Neural Networks for Efficient Language Modeling // Applied Soft Computing. 2019. Vol. 79. Pp. 354–362.
25. URL: docs.nvidia.com/deeplearning/tensorrt/best-practices/index.html#mixed-precision (дата обращения: 10.12.2020).
26. Nayak P., Zhang D., Chai S. Bit Efficient Quantization for Deep Neural Networks. URL: arxiv.org/ftp/arxiv/papers/1910/1910.04877.pdf (дата обращения: 10.12.2020).