

Разработка веб-сервиса по распознаванию знаков дорожного движения на основе сверточных нейронных сетей

К.А. Прончук¹, П.Ю. Якимов¹

¹Самарский национальный исследовательский университет имени академика С.П. Королева, Московское шоссе 34, Самара, Россия, 443086

Аннотация. Решение задачи классификации является одним из важнейших применений нейронных сетей. В настоящей статье предложен алгоритм для распознавания дорожных знаков на основе сверточных нейронных сетей. Для этого использовались современные подходы к распознаванию изображений, обучение нейронной сети происходило при помощи библиотеки TensorFlow и архитектуры параллельных вычислений CUDA. Для использования данной нейронной сети был также разработан веб-сервис на основе CUBA.platform.

1. Введение

Расширенные системы помощи водителям (ADAS) относятся к высокотехнологичным системам транспортного средства, которые предназначены для повышения безопасности дорожного движения, помогая водителям лучше понять дорогу и ее потенциальную опасность. Распознавание дорожных знаков – это одна из важных подсистем ADAS и на протяжении многих лет является сложной проблемой и, следовательно, становится важной и активной темой исследований в области интеллектуальных транспортных систем. Реализация системы распознавания дорожного знака в режиме реального времени обычно делится на два этапа: локализация дорожного знака и его классификация. Обнаружение происходит при помощи несложных вычислительных алгоритмов, таких как цветовое пороговое значение [9]. В дальнейшем классификация происходит при помощи более сложных, но в то же время более точных алгоритмов. Несмотря на то, что задача состоит исключительно в классификации, важно иметь в виду конечную цель обнаружения при разработке классификатора, чтобы оптимизировать как точность, так и эффективность.

Последнее время все большую популярность набирает такое явление как «связанные автомобили» (Connected Cars). Термин «связанные транспортные средства» относится к приложениям, услугам и технологиям, которые соединяют транспортное средство с его окружением. Обычно это означает наличие устройств в транспортном средстве, которые подключаются к другим устройствам внутри одного и того же транспортного средства и / или к устройствам, сетям, приложениям или услугам вне транспортного средства. Сервис может выполнять самые различные функции: от использования в информационно-развлекательной системе автомобиля или в системе помощи при парковке до применения в системах помощи

при вождении, а также в беспилотных автомобилях. Транспортные средства, которые включают в себя интерактивные расширенные системы помощи водителям (ADAS) и кооперативные интеллектуальные транспортные системы (C-ITS), можно рассматривать как связанные.

Как правило, компьютер на борту такого транспортного средства обладает достаточно скромными вычислительными возможностями и его использование в качестве системы помощи при вождении, а именно для распознавания дорожных знаков может быть затруднено. Однако с развитием и повсеместным распространением беспроводных технологий появилась возможность использования удаленных вычислительных мощностей для решения данной проблемы: сервис отправит изображение на удаленный сервер для обработки и дальнейшего получения ответа о наличии или отсутствии на нем дорожных знаков. Таким образом, в рамках задачи распознавания дорожных знаков, в задачи сервиса входит только передача изображения на сервер, получение ответа и отображение результата на интерактивной карте, которой воспользуются остальные транспортные средства, подключенные по принципу Connected Cars.

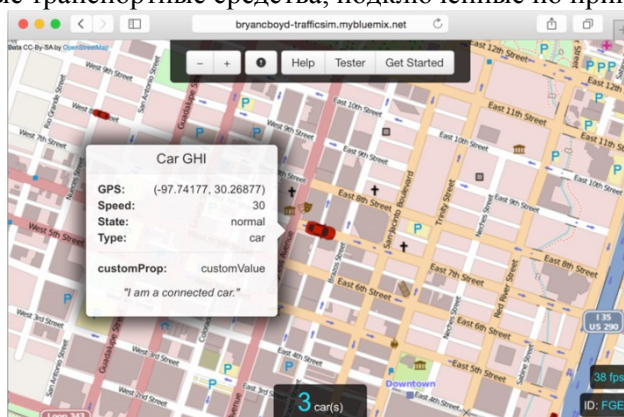


Рисунок 1. Скриншот из сервиса Connected Vehicle starter kit.

2. Метод распознавания дорожных знаков

2.1. Сверточные нейронные сети

В последнее время в задачах классификации активно используются нейронные сети. Нейронная сеть это математическая модель построенная по принципу организации и функционирования биологических нейронных сетей. Нейроны организованы в слои: входной, скрытый и выходной. Входной слой состоит не из полных нейронов, а скорее состоит из значений, которые являются входами в следующий слой. Следующий слой - скрытый. В одной нейронной сети могут существовать несколько скрытых слоев. Последний уровень - это выходной слой, где каждому классу соответствует один узел [7].

2.2. Предлагаемая реализация

Для решения поставленной задачи использовалась библиотека глубокого обучения TensorFlow. Обучение и тестирование производилось на наборе данных The German Traffic Sign Recognition Benchmark [5]. В первой версии системы распознаются 16 наиболее популярных типов дорожных знаков.

В таблице 1 представлена архитектура сверточной нейронной сети. Сеть представляет собой последовательность соединенных между собой слоев, начинающуюся со сверточного слоя и заканчивающейся слоем softmax. Параметр сверточного слоя – шаг – определяет шаг скользящего окна свертки. В случае, если шаг больше 1, сверточный слой будет совмещать в себе кроме свертки еще и операцию пулинг. Слой softmax выполняет нормализацию результатов предыдущего слоя таким образом, что на его выходе будут формироваться вероятности отношения объекта к рассматриваемым классам.

Таблица 1. Архитектура нейронной сети.

Слой
Сверточный с шагом 2 и ядром 7x7x4
Сверточный с шагом 2 и ядром 5x5x8
Сверточный с шагом 2 и ядром 3x3x16
Сверточный с шагом 2 и ядром 3x3x32
Сверточный с шагом 1 и ядром 2x2x16
Сверточный с шагом 1 и ядром 2x2x8
Сверточный с шагом 1 и ядром 2x2x4
Полносвязный с 64 выходами
Полносвязный с 16 выходами
Слой softmax

Модель можно условно поделить на 2 блока – сверточный блок и полносвязный блок. В состав фреймворка TensorFlow входят инструменты для отображения сети, которые позволяют визуализировать модель на разных уровнях абстракции, вплоть до низкоуровневых математических операций.

Для обучения модели исходный набор данных был случайно разделен на обучающую и тестовую выборки в соотношении 80 к 20. Во время обучения за одну итерацию сеть обрабатывала пачку из 50 изображений из обучающей выборки. Каждые 100 итераций подсчитывалась промежуточная точность классификации на пачке из 50 случайных изображений тестовой выборки. После завершения обучения точность была подсчитана на всей тестовой выборке. На рис. 4 продемонстрирован рост точности классификации со временем. Уже на 2000 итерации сеть достигает точности более 0.90, а остальное время происходит более точная подстройка весов [9].

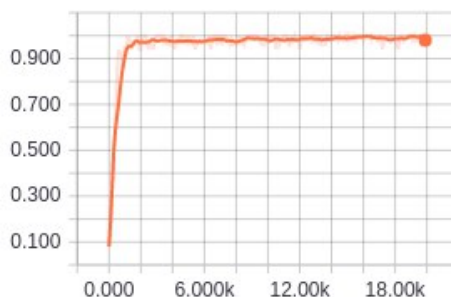


Рисунок 2. Изменение точности классификации со временем.

Но у сверточных сетей есть также и недостатки, одни из которых слой max-pooling и инвариантность детектирования лишь к положению на изображении. Сеть, обученная распознаванию типа изображения, может ошибочно распознавать один и тот же тип изображения с определенным вращением, поэтому обучение сопровождается увеличением объема тестовых данных и добавлением слоя max-pooling. Слой используется для уменьшения размерности выходов сверточных слоев, игнорирования небольших отличий в пространственной структуре изображений, при этом теряется часть информации.

Рассмотрим архитектуру капсульной нейронной сети описанной Хинтоном в [10].

Сначала идет стандартное выделение особенностей, инвариантных к трансляции на изображении, с помощью сверточного слоя. Затем выделяются еще 32 сверточных слоя, каждый из которых связан с первым сверточным слоем. Это необходимо для динамического роутинга. Далее следует слой «digitCaps» — собственно те капсулы, о которых говорит Хинтон.

Каждая капсула имеет строгий смысл: амплитуда вектора в каждой из них соответствует вероятности наличия в изображении одного из искомых классов.

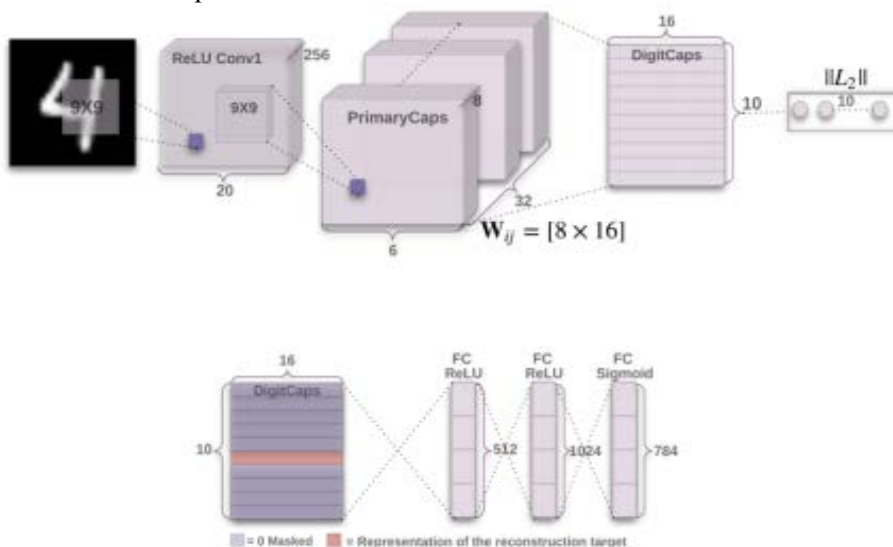


Рисунок 3. Архитектура капсульной нейронной сети.

В нижней части изображения представлена важная часть архитектуры: декодер, на вход которого подается слой DigitCaps, при этом все капсулы кроме той, у которой наибольшая амплитуда обнуляются. Декодеры — по сути инвертированная ИНС (искусственная нейронная сеть), на входе которой подали небольшой вектор, а на выходе — исходное изображение. Этот декодер обеспечивает вторую самую важную часть этой архитектуры: содержимое одной капсулы должно полностью описывать подмножество конкретного класса, подаваемое на вход. Иначе декодер никак не сможет восстановить исходное изображение. Есть и другое полезное свойство у декодера — это способ регуляризации. Близкие коды в капсулах будут у близких по Евклиду изображений.

В рамках данной работы была также обучена и использована для распознавания знаков капсульная нейронная сеть. Сеть обучалась на датасете объемом 39,209 изображений и содержащем 43 класса дорожных знаков. На рис. 5 продемонстрирован рост точности классификации со временем.

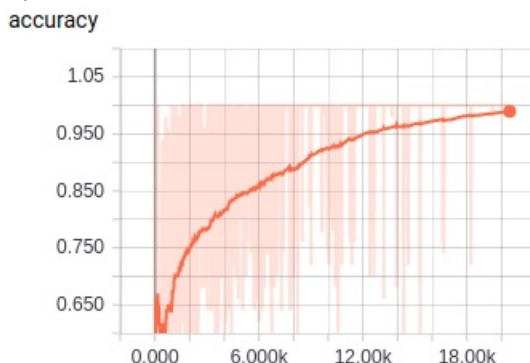


Рисунок 4. Изменение точности классификации со временем.

Капсульная нейронная сеть показала лучшую точность распознавания знаков по сравнению со сверточной сетью (при одинаковом объеме выборки и количестве классов), однако сеть обучалась несколько дольше.

3. Разработка веб-сервиса распознавания дорожных знаков

Для использования обученной нейронной сети был написан веб-сервис на платформе CUBA. CUBA — платформа для быстрой разработки бизнес-приложений на Java которая позволяет быстро создавать UI и логику для работы с данными. Для отображения графического интерфейса используется фреймворк Vaadin, который предоставляет богатый набор компонентов, возможности расширения и кастомизации визуального представления с помощью SCSS. В качестве основной инфраструктуры среднего слоя и клиентских блоков приложения используется контейнер Spring.

Для удобства использования платформы используется CUBA Studio. Это отдельное приложение, которым пользуется разработчик параллельно с обычной Java IDE. Студия предоставляет графический интерфейс к механизмам платформы, которое позволяет при помощи мыши создать модель данных, сгенерировать DDL-скрипты для БД, нарисовать экраны в WYSIWYG-редакторе, сделать заготовки сервисов среднего слоя.

Приложения на базе CUBA имеют стандартную трёхслойную архитектуру. Связующим элементом системы являются метаданные – информация о модели данных приложения. Благодаря метаданным визуальные компоненты знают о том, с какими данными они работают. Таким же образом метаданные помогают визуальным компонентам работать с базой данных через ORM, задавая графы объектов, которые нужно загрузить или обновить. Тот же принцип применяется к подсистеме безопасности, генерации отчётов и прочим частям платформы.

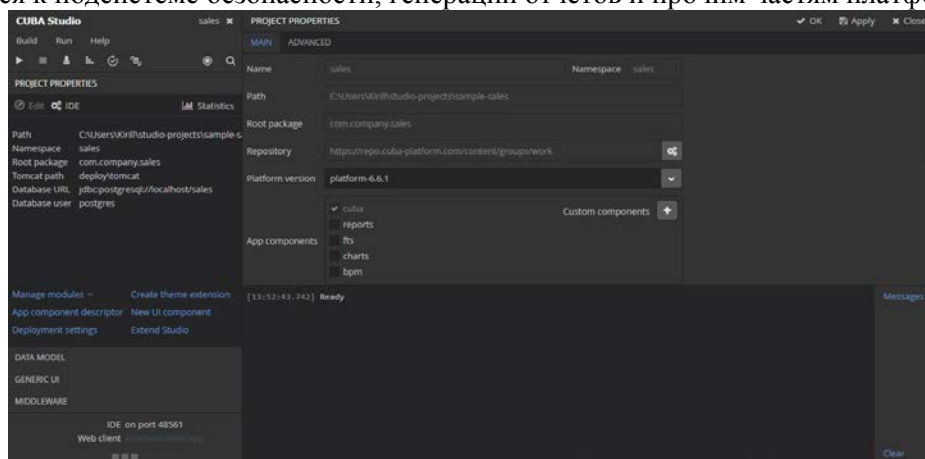


Рисунок 5. Основная информация о проекте в CUBA Studio.

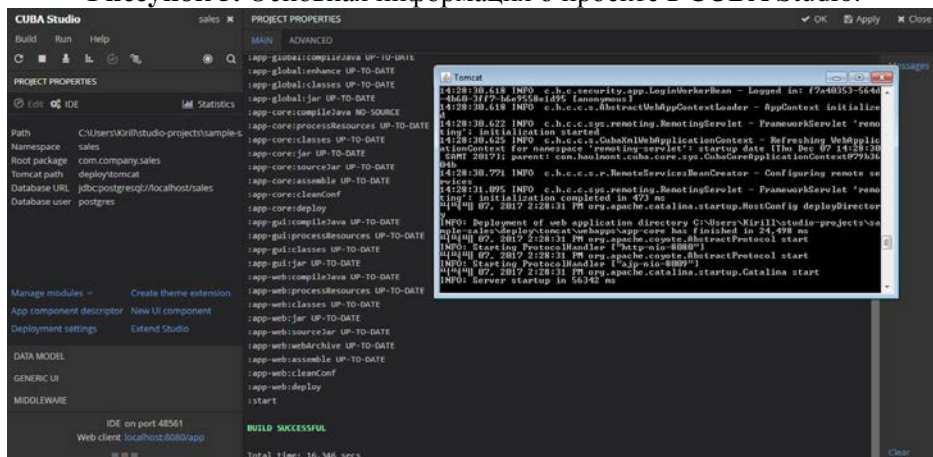


Рисунок 6. Deploy проекта в CUBA Studio.

Приложения, созданные на платформе CUBA, можно развернуть в различных конфигурациях, начиная от запуска всех компонентов приложения на одном сервере и до

конфигураций, обеспечивающих высокую отказоустойчивость, с разделёнными web, средним и БД слоями. Платформа «из коробки» поддерживает PostgreSQL, Oracle Database, Microsoft SQL Server, MySQL и HSQL (последняя обычно используется для прототипирования), между которыми можно переключаться по мере роста проекта. Также важно заметить, что CUBA-приложения могут быть развёрнуты на любом сервере Java EE Web Profile, включая Jetty, Tomcat, Glassfish, Websphere и т.д.

4. Экспериментальные исследования

Эксперимент по классификации дорожных знаков проводился одним вычислительном устройстве - процессоре Intel Core 2 quad. Сеть обучалась в течении 56 минут.

Для того, чтобы оценить точность алгоритма обнаружения, была использована немецкая база данных аннотированных изображений, содержащих дорожные знаки [5]. Она содержит более 50 000 изображений с дорожными знаками, зарегистрированными в различных условиях. Для оценки качества обнаружения было посчитано количество изображений с правильно локализованными и классифицированными дорожными знаками. При тестировании разработанных алгоритмов использовались только 39,209 изображений. Эксперименты показали 98,85% правильно локализованных и классифицированных запрещающих и предупреждающих дорожных знаков.

Среднее время обработки одного изображения составляет ~30с. Это связано с тем, что используется не Java API библиотеки Tensorflow, а вызывается Python скрипт. В дальнейшем, при использовании API предполагается значительный прирост производительности веб-сервиса.



Рисунок 7. Примеры распознанных дорожных знаков.

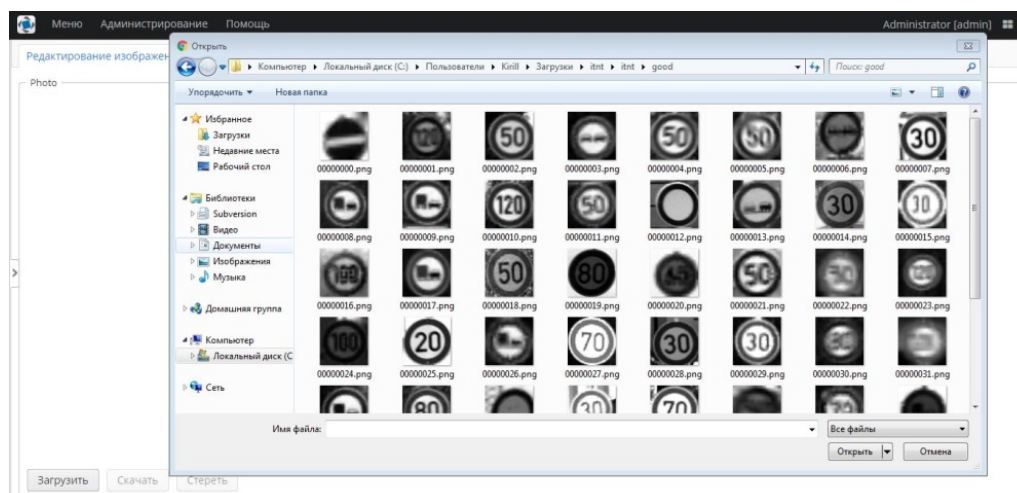


Рисунок 8. Внешний вид веб-сервиса.

5. Заключение

В процессе исследований были получены следующие результаты: разработан метод классификации дорожных знаков при помощи капсульных нейронных сетей, реализована серверная часть сервиса по распознаванию дорожных знаков.

Экспериментальные исследования показали высокую эффективность классификации – 98,85%. При этом капсульные нейронные сети продемонстрировали лучший результат по сравнению со сверточными нейронными сетями. Это стало возможно благодаря более сложному устройству CapsNet, однако при этом также уменьшилась производительность.

Разработанный веб-сервис предназначен для использования в концепции Connected Car. Такой сервис позволяет создавать и заполнять интерактивные карты, сообщать другим ТС о дорожных событиях, а в дальнейшем может быть использован и для беспилотного автомобиля.

Основным недостатком предложенной системой является большое время, затрачиваемое на вызов скрипта. В дальнейших исследованиях планируется доработать и расширить набор REST запросов веб сервиса, а также использовать Tensorflow Java API для загрузки обученной сети вместо вызова Python скрипта, что существенно сократит время обработки каждого отдельного изображения.

6. Благодарности

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-37-60106 мол_а_дк.

7. Литература

- [1] Shneier, M. Road sign detection and recognition / M. Shneier // Proc. IEEE Computer Society Int. Conf. on Computer Vision and Pattern Recognition. – 2005. – P. 215-222.
- [2] Ruta, A. A New Approach for In-Vehicle Camera Traffic Sign Detection and Recognition / A. Ruta, F. Porikli, Y. Li, S. Watanabe, H. Kage, K. Sumi // IAPR Conference on Machine Vision Applications (MVA), Session 15: Machine Vision for Transportation, 2005.
- [3] Belaroussi, R. Road Sign Detection in Images / R. Belaroussi, P. Foucher, J.P. Tarel, B. Soheilian, P. Charbonnier, N. Paparoditis // A Case Study, 20th International Conference on Pattern Recognition (ICPR). – 2010. – P. 484-488.
- [4] Якимов, П.Ю. Отслеживание дорожных знаков в видеопоследовательности с использованием скорости автомобиля / П.Ю. Якимов // Компьютерная оптика. – 2015. – Т. 39, № 5. – С. 795-800.

- [5] Houben, S. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark / S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel // International Joint Conference on Neural Networks, 2013.
- [6] Zhu, Z. Traffic-Sign Detection and Classification in the Wild / Z.e Zhu, D. Liang, S. Zhang, X. Huang, B. Li, Sh. Hu // Proceedings of CVPR. – 2016. – P. 2110-2118.
- [7] LeCun, Y. Traffic Sign Recognition with Multi-Scale Convolutional Networks / Y. LeCun, P. Sermanet // Proceedings of International Joint Conference on Neural Networks (IJCNN'11), 2011.
- [8] Mathias, M. Traffic sign recognition - how far are we from the solution? / M. Mathias, R. Timofte, R. Benenson, L. V. Gool // Proceedings of IEEE International Joint Conference on Neural Networks. – 2013. – P. 1-8.
- [9] Якимов, П.Ю. Использование ГПУ в задаче классификации дорожных знаков при помощи сверточных нейронных сетей / П.Ю. Якимов, А.В. Шустанов // Информационные технологии и нанотехнологии (ИТНТ-2017). – 2017. – С. 1691-1696.
- [10] Hinton, G.E. Dynamic Routing Between Capsules / G.E. Hinton, S. Sabour, N. Frosst // Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2017), 2017.
- [11] Koesdwiady, A. Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach / A. Koesdwiady, R. Souza, F. Karray // Proceedings of IEEE Transactions on Vehicular Technology. – 2016. – Vol. 65(12). – P. 9508-9517.

Web service development for road signs recognition based on convolutional neural networks

K. Pronchuk¹, P. Yakimov¹

¹Samara National Research University, Moskovskoe Shosse, 34A, Samara, Russia, 443086

Abstract. The one of the most general application domain of neural networks is a problem of pattern recognition. In this paper we present road signs recognition algorithm based on convolutional neural networks. For that aim we use modern approaches of pattern recognition, neural network was trained using machine learning software library TensorFlow and parallel computing platform CUDA. For later use of this neural network we develop a web-service based on CUBA.platform.

Keywords: convolutional neural networks, deep learning, tensorflow, web-service, road signs recognition.