

ТЕХНИЧЕСКИЕ НАУКИ

Научная статья

УДК 004.4:004.032.26

doi: 10.34822/1999-7604-2022-2-6-13

ТЕСТИРОВАНИЕ И АНАЛИЗ ФРЕЙМВОРКОВ, ПРЕДНАЗНАЧЕННЫХ ДЛЯ ГЛУБОКОГО ОБУЧЕНИЯ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

**Сергей Юрьевич Пчелинцев^{1✉}, Ольга Александровна Ковалева²,
Андрей Алексеевич Суслин³**

^{1,2,3}Тамбовский государственный университет им. Г. Р. Державина, Тамбов, Россия

²Тамбовский государственный технический университет, Тамбов, Россия

¹veselyrojer@mail.ru[✉], <http://orcid.org/0000-0001-9195-8318>

²solomina-oa@yandex.ru, <http://orcid.org/0000-0003-0735-6205>

³andriw_suslin@mail.ru, <http://orcid.org/0000-0003-2219-4013>

Аннотация. В статье приведено сравнение нескольких фреймворков глубокого обучения в формате тестирования в схожих условиях с целью их дальнейшего применения для решения задач в режиме реального времени на различных платформах. Одним из примеров такой задачи является определение дорожных знаков. В исследовании проводится сравнение скорости и точности выходных данных пяти популярных фреймворков глубокого обучения: TensorFlow, Neon, CNTK, MXNet, PyTorch. Для обучения в условиях ограниченных вычислительных ресурсов используются три наиболее популярные модели обучения (ResNet-20, IDSIA и ResNet-32), выбранные из-за высокоточных результатов в режиме реального времени. Именно графические процессоры обычно используются для обучения сверточных нейронных сетей. В статье собраны и проанализированы результаты вычислений с использованием как центральных, так и графических процессоров. Представлены методы оптимизации работы фреймворков, результаты их работы как на центральном, так и на графическом процессоре. Благодаря этому появилась возможность исследовать влияние входных данных на точность получаемого впоследствии результата. Данное исследование дает возможность определить и оптимизировать наиболее подходящий по критериям точности выводимых значений и затрачиваемого для обучения моделей времени фреймворк для распознавания дорожных знаков.

Ключевые слова: нейросетевые модели, глубокое обучение, фреймворк

Для цитирования: Пчелинцев С. Ю., Ковалева О. А., Суслин А. А. Тестирование и анализ фреймворков, предназначенных для глубокого обучения нейросетевых моделей // Вестник кибернетики. 2022. № 2 (46). С. 6–13. DOI 10.34822/1999-7604-2022-2-6-13.

Original article

TESTS AND ANALYSIS OF FRAMEWORKS INTENDED FOR DEEP LEARNING OF NEURAL NETWORK MODELS

Sergey Yu. Pchelintsev^{1✉}, Olga A. Kovaleva², Andrey A. Suslin³

^{1,2,3}Derzhavin Tambov State University, Tambov, Russia

²Tambov State Technical University, Tambov, Russia

¹veselyrojer@mail.ru[✉], <http://orcid.org/0000-0001-9195-8318>

²solomina-oa@yandex.ru, <http://orcid.org/0000-0003-0735-6205>

³andriw_suslin@mail.ru, <http://orcid.org/0000-0003-2219-4013>

Abstract. The article compares several deep learning frameworks via testing under similar conditions in order to apply them in solving real-time problems on various platforms. Traffic signs recognition is an example of such problem. The study compares speed and accuracy of data output of five popular deep learning frameworks: TensorFlow, Neon, CNTK, MXNet, PyTorch. The most popular training models, ResNet-20, IDSIA, ResNet-32, were selected for training in conditions of limited computing resources due to their highly accurate real-time results. Graphics processing units are commonly used to train convolutional neural networks. The article compiled and analyzed the results of calculations using both central and graphics processors, presented methods of frameworks optimization and the results of their operation both on the central processor and on the graphics one. Thus, it is possible to study the data input influence on the accuracy of subsequently obtained result. The study allows determining and optimizing the most suitable framework for traffic signs recognition according to the criteria of output values accuracy and the time spent on models training.

Keywords: neural network models, deep learning, framework

For citation: Pchelintsev S. Yu., Kovaleva O. A., Suslin A. A. Tests and Analysis of Frameworks Intended for Deep Learning of Neural Network Models // *Proceedings in Cybernetics*. 2022. No. 2 (46). P. 6–13. DOI 10.34822/1999-7604-2022-2-6-13.

ВВЕДЕНИЕ

Метод глубокого обучения, включающий в себя глубокие нейросетевые модели, решает одну из важнейших проблем ограниченности вычислительных ресурсов встроённых систем, которая не позволяет в короткое время получать высокоточные выходные результаты. В свою очередь, решение такой проблемы является очень важным фактором для создания самоуправляемых автомобилей, от обучения которых зависит безопасность водителей, пассажиров и пешеходов.

Для достижения вышеупомянутых высокоскоростных и высокоточных данных требуется использовать механизм параллельных вычислений [1]. В наше время существует большое количество фреймворков, дающих возможность заниматься глубоким обучением нейросетевых моделей при помощи параллелизма. Для того чтобы выбрать наиболее подходящие фреймворки, необходимо произвести их сравнение в различных условиях работы.

Новизна исследования заключается в применении сравнительного анализа фреймворков, проведенного по трем свёрточным нейросетевым моделям с использованием входных данных размера 48×48 .

Одной из проблем свёрточных нейронных сетей является то, что после завершения обучения [2–3] и определения сети размер входных изображений должен быть фиксированным, но в реальных приложениях до этапа классификации существует возможность за-

грузки данных с минимальным и максимальным размером. Очень важно определиться с размером входного изображения нейронной сети по нескольким причинам:

1. Конечным пользователям зачастую неизвестна низкоуровневая оптимизация вычислений фреймворка.
2. Из-за ограниченности доступных ресурсов может быть недоступна обработка большого изображения.
3. От размера входных данных может зависеть точность выходных данных.

Помимо всего перечисленного, важно минимизировать время, затрачиваемое на обучение модели. Поэтому в исследовании большое внимание будет уделено размерам входных данных, производительности и точности выходных данных.

МАТЕРИАЛЫ И МЕТОДЫ

В качестве сравниваемых фреймворков были отобраны пять наиболее распространённых: CNTK2.7.0, Tensorflow2.7.0, PyTorch1.3, Neon2.6.0 и MXNet1.5.0. В ходе исследования используется популярный набор данных, в том числе содержащий более 50 тыс. изображений дорожных знаков немецкий бенчмарк GTSRB [4]. Изображения представлены в RGB-, Haar-, HueHist- и HOG-представлениях, а их размер варьируется от 15×15 до 250×250 пикселей.

Также использовались три модели нейронных сетей IDSIA, ResNet-20 и ResNet-32 (две последние – свёрточные нейронные сети).

Эти модели демонстрируют высокую точность получаемого результата и требуют сравнительно небольшого объема входных данных [5].

Модель IDSIA, обучаемая в ходе исследования, показала высокую точность классифи-

кации – 99 %. В результате применения предварительной обработки данных сверточные нейронные сети способны демонстрировать высокую точность, даже в случае, когда есть значительное количество слоев [6]. Схема модели IDSIA представлена на рис. 1.

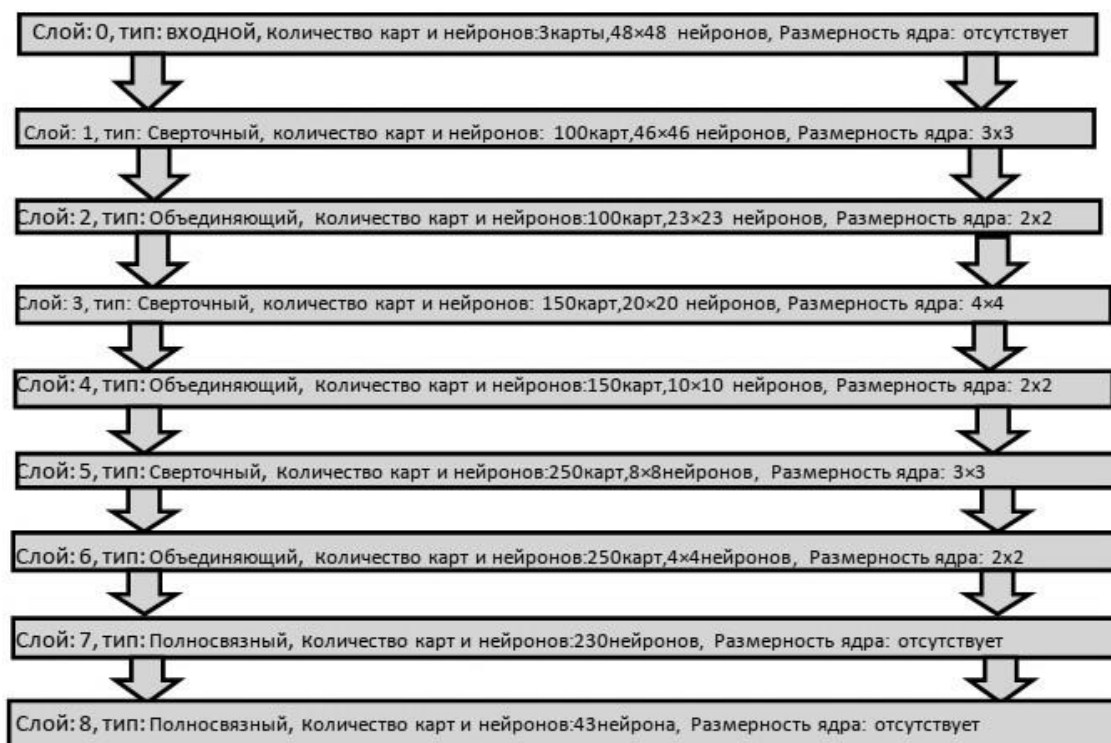


Рис. 1. Структура модели IDSIA

Примечание: составлено авторами на основании материалов из открытых источников.

В исследованиях использовалась современная модель ResNet, которая предназначена для снятия ограниченности глубины со сверточных сетей и решения проблемы взрывного роста градиента, вследствие чего выходные размеры оказываются гибкими, но

при этом ResNet имеет сравнительно малое количество параметров. К примеру, ResNet-32 имеет только 0,46 млн параметров, в то время как AlexNet и VGG имеют 60 и 138 млн соответственно [7]. Структура ResNet представлена на рис. 2.

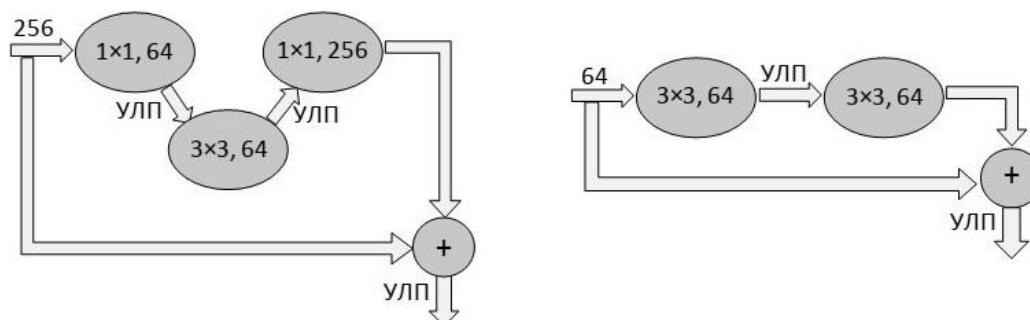


Рис. 2. Основные блоки построения нейросетевой модели ResNet, где УЛП – это усеченное линейное преобразование

Примечание: составлено авторами на основании материалов из открытых источников.

Зачастую изображения, содержащиеся в наборе обучающих данных, имеют как различия в размере, так и разницу в освещении. Поэтому для повышения качества обучения сверточных нейронных сетей [8] и профилактики ошибок требуется предварительная обработка используемых изображений. Для обработки можно применить выравнивание гистограммы и адаптивное выравнивание с ограниченной контрастностью (CLAHE). Вначале изображения обрезаются, а затем приводятся к размерам 48×48 для сетей IDSIA и ResNet.

В модели ResNet общее количество слоев определяется как $6n + 2$ (n – это нечетное число). В процессе построения нейросетевых моделей ResNet-20 и ResNet-32 использовались $n = 3$ и $n = 5$ соответственно. Причиной тому является дальнейшее уменьшение изображения для карты признаков в модели ResNet вдвое; следовательно, размер входного сигнала должен быть кратен 4. Поэтому практичнее всего для входных изображений (после изменения размеров) использовать в качестве размеров следующие значения: 32, 48 и 64. В эксперименте используются изображения 48×48 .

Эксперименты проводились на компьютере, оснащенный следующими компонентами: графическим ускорителем GTX 1 060 с оперативной памятью, равной 6 Гб; процессором Intel Core I5 4570 на 3.2 Гц, 24 Гб оперативной памяти. Имеющийся код был написан на Python 3.6 с использованием встраиваемых и стандартных библиотек.

В процессе эксперимента применяется алгоритм стохастического градиентного спуска, шаг обучения которого равен 0,01. Для оптимизатора норму изменений берем равной 0,9. При этом число эпох равно 25. Размер пакета данных составляет 64. Инициализация сверточных слоев осуществляется при помощи `he_normal`, который обеспечивает лучшую сходимость. Оптимизация используемых фреймворков производится индивидуально при помощи сборки из исходного кода и настройки при помощи Python.

В целях ускорения исследования каждый фреймворк тестируется в неизменных усло-

виях, то есть с одной эпохой обучения при помощи модели IDSIA в сочетании с набором данных GTSRB, равным 31 367 изображениям. Размер входных данных 48×48 , а размер пакета данных – 64. Производительность фреймворков с использованием как процессора, так и графического ускорителя увеличивается при помощи системной оптимизации и оптимизации кода.

Скорость работы фреймворков (MXNet, Neon и TensorFlow) на центральном процессоре перед сборкой можно повысить следующим образом:

- для MXNet устанавливаются флаги `USE_MKL_EXPERIMENTAL=1` и `USE_MKL=1`, которые повышают скорость его работы на процессоре до 114,4 секунд на эпоху (в сравнении с первоначальной скоростью 219,9 секунд);
- Neon можно оптимизировать путем установки определенного количества физических ядер локальной машины (в данной ситуации 4) для `OMP_NUM_THREADS`, а также установкой значений `compact: 1, 0` – для `KMP_AFFINITY`;
- скорость обучения TensorFlow увеличивается при помощи установки пакета Python, созданного TinyMind.

Оставшиеся же CNTK и PyTorch, поставляемые с поддержкой MKL, в поддержке такого рода не нуждаются.

Стоит отметить, что есть исключения и не все вышеуказанные методы воздействия и оптимизации фреймворков действуют на различные версии одинаково. Существует возможность повысить скорость и точность фреймворков путем оптимизирования кода. Для CNTK и других фреймворков достаточно установить значения указанных параметров `use_mean_gradient=True` и `unit_gain=False`, такое решение повысит производительность и решит некоторые проблемы при обучении сетей.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

На рис. 3–4 представлено время обучения трех моделей при фиксированном размере входных данных (48×48).

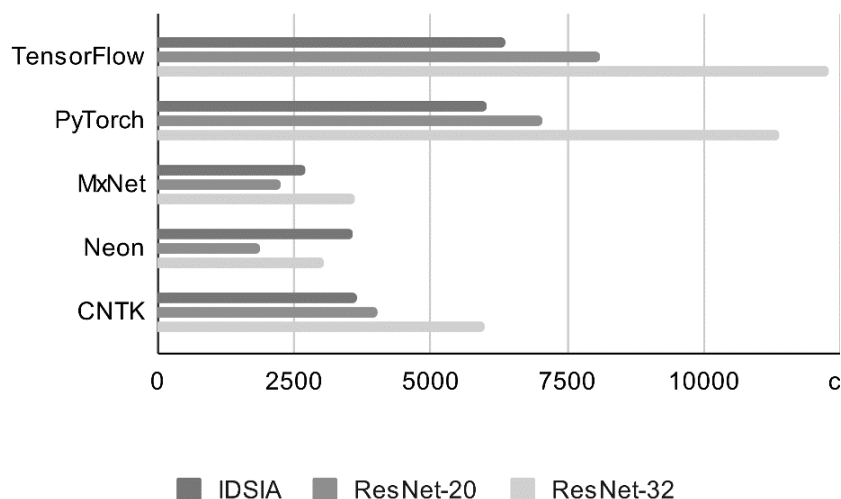


Рис. 3. Время обучения трех нейросетевых моделей на графическом процессоре в случае размера входных данных 48×48
 Примечание: составлено авторами на основании данных, полученных в исследовании.

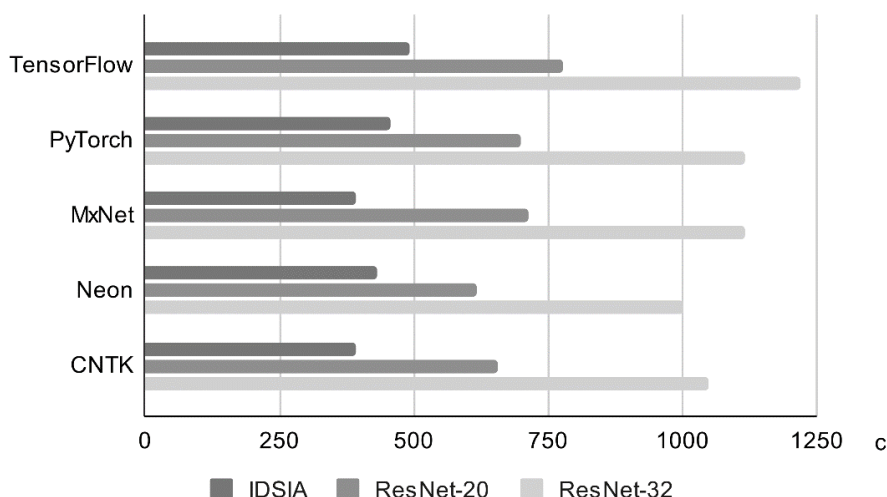


Рис. 4. Время обучения трех нейросетевых моделей на центральном процессоре в случае размера входных данных 48×48
 Примечание: составлено авторами на основании данных, полученных в исследовании.

В случае использования графического процессора полученные результаты свидетельствуют о том, что среди всех испытуемых самую высокую скорость обучения демонстрирует Neon. Имеется лишь одно исключение: MXNet опережает Neon при работе с IDSIA с размером входных изображений, равным 48×48 . В целом скорость обучения всех фреймворков приблизительно одинакова.

По скорости обучения первые места занимают Neon и MXNet при использовании центрального процессора. Важно отметить, что обучение фреймворков MXNet и Neon для модели нейронной сети IDSIA заняло

больше времени, чем для более глубокой нейросетевой модели ResNet-20. Причиной могут являться массивные вычисления, производимые данными фреймворками на центральном процессоре, которые не оптимизированы так же эффективно, как на графическом. Подтверждением этому является то, что остальные фреймворки при обучении на центральном процессоре показали результат хуже, чем на графическом.

Абсолютными фаворитами по точности выходных данных являются Neon и TensorFlow (рис. 5–6). В то же время разница между точностью выходных данных среди

всех фреймворков незначительна. Стоит отметить, что, как и предполагалось, две модели ResNet в значительной степени превосхо-

дят IDSIA, но в то же время между собой в точности они не различаются.

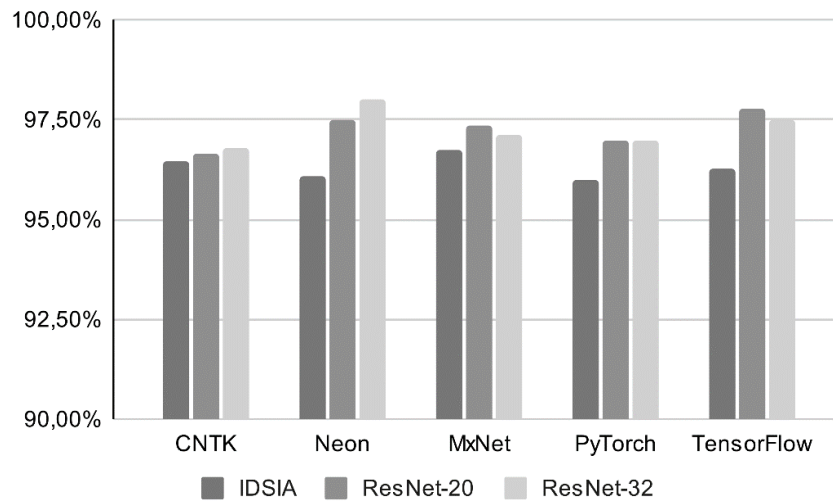


Рис. 5. Точность выходных данных трех моделей при фиксированном размере входных данных (48×48) на графическом процессоре
Примечание: составлено авторами на основании данных, полученных в исследовании.

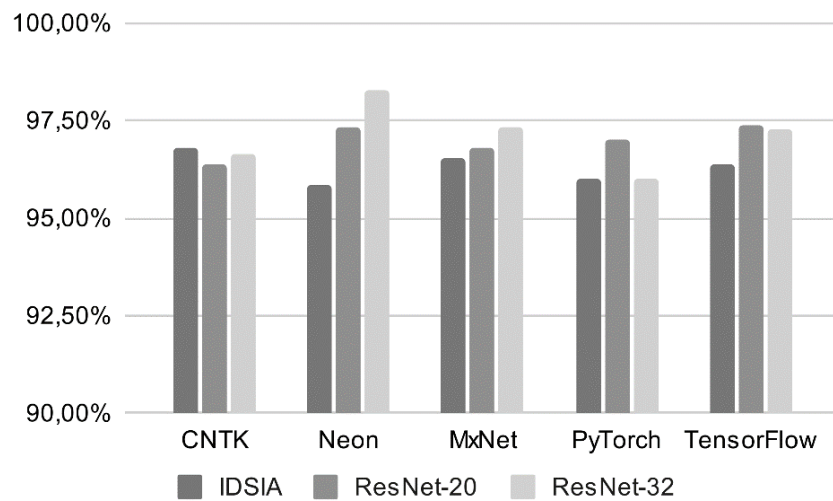


Рис. 6. Точность выходных данных трех моделей при фиксированном размере входных данных (48×48) на центральном процессоре
Примечание: составлено авторами на основании данных, полученных в исследовании.

Анализируя полученные результаты, можно сделать вывод, что самым быстрым в обучении и самым точным среди вышеупомянутых фреймворков является Neon. Сравниться с ним в точности могут лишь MXNet и TensorFlow: скорость их обучения значительно выше. Данное сравнение производится по средним результатам проведенных экспериментов. Если сравнивать результаты фреймворков лишь при использовании

графического процессора, то они будут примерно одинаковы.

По результатам проведенных экспериментов видно, что обе модели ResNet преобладают над IDSIA, но между собой большой разницы в результатах не имеют. Из этого следует, что при использовании набора данных GTSRB и ему подобных применение более глубоких, чем ResNet-20, моделей не имеет смысла, так как величина затрачивае-

мого на обучение времени не будет соизмерима с выигрышем в точности выходящих данных. Также стоит отметить, что изменение размера входных изображений при использовании набора данных GTSRB практически не влияет на точность выходных данных и масштабируемость обучающей среды.

Полученные результаты тестирования фреймворков при использовании одинаковой архитектуры самой модели, а также одинаковых размеров входных данных будут различны на центральном и графическом процессоре. Причиной может являться то, что процесс обучения на центральном процессоре детерминированный (в отличие от обучения на графическом процессоре). Это, в свою очередь, означает, что идентичные, предварительно обученные на центральном процессоре модели могут быть сгенерированы для разных устройств, но имеется исключение. CNTK путем форсирования детерминированных алгоритмов сможет воспроизвести на начальных 20–30 эпохах результаты обучения, но при этом в дальнейшем значения потерь отклоняются. При этом стоит отметить, что форсирование детерминированных алгоритмов приводит к увеличению времени обучения одной эпохи на графическом процессоре практически вдвое.

ЗАКЛЮЧЕНИЕ

Таким образом, фаворитами исследования, имеющими самую высокую точность вывода и скорость обучения, являются Neon и MXNet. Помимо перечисленных фреймвор-

ков, по степени точности как при работе на графическом процессоре, так и на центральном выделился TensorFlow. В связи с тем, что размер входных данных не всегда влияет на точность результата, при использовании набора данных GTSRB предлагается использовать значение 32×32 в качестве размера входных данных для определения дорожных знаков. Такое решение является наиболее экономичным и практически не влияет на точность результатов. Проведенные эксперименты показали, что модель ResNet-32 имеет схожую с ResNet-20 точность. Это доказывает нам, что для определения дорожных знаков более глубокие модели, имеющие более длительное время обучения, не всегда предпочтительнее малых.

Следующей стадией исследования является развертывание на различных устройствах моделей обработки данных в реальном времени, анализ получаемых результатов, а также использование более обширных, чем GTSRB, наборов данных в целях улучшения качества обучения, что приведет к повышению качества работы фреймворков, но в то же время станет катализатором новых проблем обработки изображений, которые потребуют решения. В настоящее время существует множество алгоритмов обработки данных, использование которых вкупе с реализациями, встроенными в фреймворки, в дальнейшем позволит обрабатывать большой объем данных о дорожных знаках.

Список источников

1. Чижова И. А., Некрасова А. С. Актуальность параллельных вычислений на Cuda // Наука молодых – будущее России : сб. науч. ст. 3-й Междунар. науч. конф. перспективных разработок молод. ученых, 11–12 декабря 2018 г., г. Курск. Курск : ЗАО «Университет. книга», 2018. С. 289–291.
2. Багаев И. И. Анализ понятий нейронная сеть и сверточная нейронная сеть, обучение сверточной нейросети при помощи модуля TensorFlow // Математ. и програм. обеспечение систем в промышлен. и социал. сферах. 2020. № 1. С. 15–22.
3. Ильин Е. С., Тимохов Д. В., Бабенко В. Г. Идентификация объектов при помощи сверточных нейронных сетей // Транссибирская магистраль – экономический пояс России XX : межвузов.

References

1. Chizhova I. A., Nekrasova A. S. Aktualnost parallelnykh vychislenii na Cuda // Nauka molodykh – budushchee Rossii : Proceedings of the 3rd International Research Conference on Prospect Developments of Young Scientists, December 11–12, 2018, Kursk. Kursk : ZAO “Universitet. kniga”, 2018. P. 289–291. (In Russian).
2. Bagaev I. I. Analiz poniatii neironnaia set i svertochnaia neironnaia set, obuchenie svertochnoi neiroseti pri pomoshchi modulua TensorFlow // Software of Systems in the Industrial and Social Fields. 2020. No. 1. P. 15–22. (In Russian).
3. Ilyin E. S., Timokhov D. V., Babenko V. G. Identifikatsiia obektov pri pomoshchi svertochnykh neironnykh setei // Transsibskaiia magistral – ekonomicheskii poias Rossii XX : Proceedings of

- науч.-практ. студен. конф., 18–22 апреля 2016 г., г. Красноярск. Красноярск : Краснояр. ин-т железнодорож. транспорта, 2016. С. 154–158.
4. German Traffic Sign Benchmarks. Dataset. URL: https://benchmark.ini.rub.de/gtsrb_dataset.html (дата обращения: 11.01.2022).
 5. Лобанов М. Г., Шоломов Д. Л. Об ускорении архитектуры сверточной нейронной сети на базе resnet в задаче распознавания объектов дорожной сцены // Информац. технологии и вычислит. системы. 2019. № 3. С. 57–65.
 6. Cireşan D., Meier U., Masci J., Schmidhuber J. Multi-Column Deep Neural Network for Traffic Sign Classification // Neural Netw. 2012. Vol. 32. P. 333–338.
 7. Koonce B. Convolutional Neural Networks with Swift for Tensorflow. Berkeley, CA : Apress, 2021. 245 p.
 8. Новосельцев В. А., Тушев А. Н. Исследование влияния предварительной обработки изображений на скорость обучения и качество распознавания нейронной сети // Программно-технич. обеспечение автоматизир. систем : материалы Всерос. молодеж. науч.-практ. конф., 9 ноября 2018 г., г. Барнаул. Барнаул : Алтайский гос. техн. ун-т им. И.И. Ползунова, 2018. С. 175–182.
 - the Inter-Universities Research-to-Practice Student Conference, April 18–22, 2016, Krasnoyarsk. Krasnoyarsk : Krasnoyarsk Railway Transport Institute, 2016. P. 154–158. (In Russian).
 4. German Traffic Sign Benchmarks. Dataset. URL: https://benchmark.ini.rub.de/gtsrb_dataset.html (accessed: 11.01.2022).
 5. Lobanov M. G., Sholomov D. L. On the Acceleration of the Convolutional Neural Network Architecture Based on Resnet in the Task of Road Scene Objects Recognition // Journal of Information Technologies and Computing Systems. 2019. No. 3. P. 57–65. (In Russian).
 6. Cireşan D., Meier U., Masci J., Schmidhuber J. Multi-Column Deep Neural Network for Traffic Sign Classification // Neural Netw. 2012. Vol. 32. P. 333–338.
 7. Koonce B. Convolutional Neural Networks with Swift for Tensorflow. Berkeley, CA : Apress, 2021. 245 p.
 8. Novoseltsev V. A., Tushev A. N. Issledovanie vliianiia predvaritelnoi obrabotki izobrazhenii na skorost obucheniia i kachestvo raspoznavaniia neironnoi seti // Programmno-tekhnich. obespechenie avtomatizir. sistem : Proceedings of the All-Russian Youth Research-to-Practice Conference, November 9, 2018, Barnaul. Barnaul : Polzunov Altay State Technical University, 2018. P. 175–182. (In Russian).

Информация об авторах

С. Ю. Пчелинцев – аспирант.
О. А. Ковалева – доктор технических наук, доцент.
А. А. Суслин – студент.

Information about the authors

S. Yu. Pchelintsev – Postgraduate.
O. A. Kovaleva – Doctor of Sciences (Engineering), Associate Professor.
A. A. Suslin – Student.