СТАТЬЯ

УДК 004.942

НЕЙРОСЕТЕВАЯ МОДЕЛЬ ДЛЯ РАСПОЗНАВАНИЯ ДОРОЖНЫХ ЗНАКОВ

Смольянинов В.А., Гришунов С.С., Белов Ю.С.

Московский государственный технический университет имени Н.Э. Баумана, Калужский филиал, Калуга, e-mail: fn1-kf@mail.ru

В данной статье описывается процедура обучения модели сверточной нейронной сети для распознавания дорожных знаков на изображениях. Модель была спроектирована на основе глубокой сверточной нейронной сети для распознавания рукописных цифр из исследования А. Шона и соавторов, а также модели из исследования Rinu Gour. Для обучения модели был сгенерирован набор изображений на основе российской базы автодорожных знаков RTSD. Сгенерированный набор данных включает 53 различных класса дорожных знаков и насчитывает более 40 тыс. изображений 40×40 пикселей. В процессе обучения модели используется предварительная обработка размытых и зашумленных изображений, а также аугментация данных, что, в свою очередь, позволяет улучшить точность предсказания. Для реализации и обучения модели использовались библиотеки глубокого обучения TensorFlow и Keras совместно с библиотеками Cuda и Cudnn для ускоренного обучения сети на GPU. Модель обучалась с использованием системы на базе процессора AMD Ryzen 5 2600 Six-Core Processor 3.4 GHz, 16 GB оперативной памяти класса DDR4 3200 MHz и графического процессора NVIDIA GeForce GTX 1660 6GB GDDR5 с частотой ядра/памяти: 1830/8000 MHz соответственно. Время обучения модели на GPU заняло 15,5 мин, а время обучения на CPU составило 4,2 ч для 30 эпох. Точность модели составила 99,22%.

Ключевые слова: распознавание дорожных знаков, сверточная нейронная сеть, обучение нейросетевой модели, аугментация данных

NEURAL NETWORK MODEL FOR TRAFFIC SIGN RECOGNITION

Smolyaninov V.A., Grishunov S.S., Belov Yu.S.

Bauman Moscow State Technical University, Kaluga branch, Kaluga e-mail: fn1-kf@mail.ru

This article describes the procedure for training a convolutional neural network model for recognizing traffic signs in images. The model was designed on the basis of a deep convolutional neural network for handwritten digit recognition from the study by A. Shawon and co-authors, as well as the model from the Rinu Gour study. To train the model, a set of images was generated based on the Russian RTSD database of traffic signs. The generated dataset includes 53 different classes of traffic signs and contains more than 40 thousand images of 40×40 pixels. In the process of training the model, preliminary processing of blurry and noisy images is used, as well as data augmentation, which, in turn, improves the prediction accuracy. To implement and train the model, the TensorFlow and Keras deep learning libraries were used together with the Cuda and Cudnn libraries for accelerated network training on the GPU. The model was trained using a system based on an AMD Ryzen 5 2600 Six-Core Processor 3.4 GHz, 16 GB of DDR4 3200 MHz RAM and an NVIDIA GeForce GTX 1660 6GB GDDR5 graphics processor with a core / memory frequency of 1830/8000 MHz, respectively. The GPU training time was 15.5 minutes, while the CPU training time was 4.2 hours for 30 epochs. The accuracy of the model was 99.22%.

Keywords: traffic sign recognition, convolutional neural network, neural network model training, data augmentation

В предыдущей статье [1] на основе материалов исследования различных подходов и методов обработки изображений, содержащих объекты заданного класса [2] была спроектирована модель нейронной сети для распознавания дорожных знаков. Эта модель представляет собой общий иерархический экстрактор признаков, который преобразует обработанные интенсивности пикселя входного изображения в вектор признаков, который классифицируется двумя полносвязными слоями. В данной статье рассматривается процедура обучения модели на наборе данных, сгенерированном на основе российской базы автодорожных знаков RTSD [3].

Цель исследования: рассмотреть процесс обучения сверточной нейронной сети, сформировать обучающий набор данных и усовершенствовать модель распознавания дорожных знаков, а также проанализировать полученную модель и зафиксировать результаты работы.

Формирование набора данных. Для проведения эксперимента использовался набор данных, сгенерированный на основе российской базы автодорожных знаков RTSD. Для составления базы RTSD использовались кадры, предоставленные компанией Геоцентр-Консалтинг [3]. Кадры получены с широкоформатных видеорегистраторов, установленных за лобовым стеклом автомобиля. Видеорегистраторы снимают со скоростью 5 кадров в секунду. Разрешение кадров варьируется от 1280×720 до 1920×1080. Также стоит отметить, что кадры сняты в различные времена суток (утро, день, вечер), времена года (зима, весна, лето, осень), а также при различных погодных условиях (яркое солнце, дождь, снег). База RTSD содержит 156 различных знаков и 104358 изображений дорожных знаков.

Рис. 1. Набор изображений 40×40 для каждого из 53 классов

Набор данных для обучения предлагаемой модели состоит из 42689 изображений с 53 различными классами. Изображения распределены между этими классами неравномерно, поэтому модель может предсказывать некоторые классы с более высокой точностью, чем другие. Поэтому целесообразно сделать набор данных для обучения более согласованным, увеличив выборку данных путем модификации существующих изображений. Такой метод называется аугментацией данных.

Одно из ограничений модели сверточной нейронной сети состоит в том, что она не может быть обучена на изображениях разного размера. Таким образом, в наборе данных обязательно должны быть изображения одного и того же размера.

В исходном наборе данных RTSD-3 изображения имеют динамический диапазон измерений, варьирующийся от 16×16×3 до 128×128×3, следовательно, он не может быть передан непосредственно в модель. С помощью библиотеки OpenCV методом cv2.resize(img, (40, 40)) была установлена размерность (40×40) всех изображений в наборе данных (рис. 1).

Обучение сверточной нейронной сети. Имеющийся набор данных предварительно обрабатывается перед началом обучения, а затем, во время обучения, случайным образом искажается перед началом каждой следующей эпохи. Стоит заметить, что предварительная обработка не является стохастической и выполняется для всего набора данных до начала обучения. Однако искажения являются стохастическими и применяются к каждому предварительно обработанному изображению во время обучения, используя случайные, но ограниченные значения трансляции, поворота и масштабирования. Эти значения взяты из равномерного

распределения в заданном диапазоне, т.е. ±10% от размера изображения для сдвига (параметры width_shift_range и height_shift_range), 0,8–1,2 – для масштабирования (параметр zoom_range) и ±10 градусов – для вращения (параметры rotation_range и shere_range). Алгоритм учитывает только те классы, в которых менее чем 1000 изображений, и выполняет одну из операций преобразования. Полученные изображения добавляются в один и тот же класс до тех пор, пока число его элементов не достигнет 1000 единиц [4]. Процедура обучения сверточной нейронной сети показана на рис. 2.

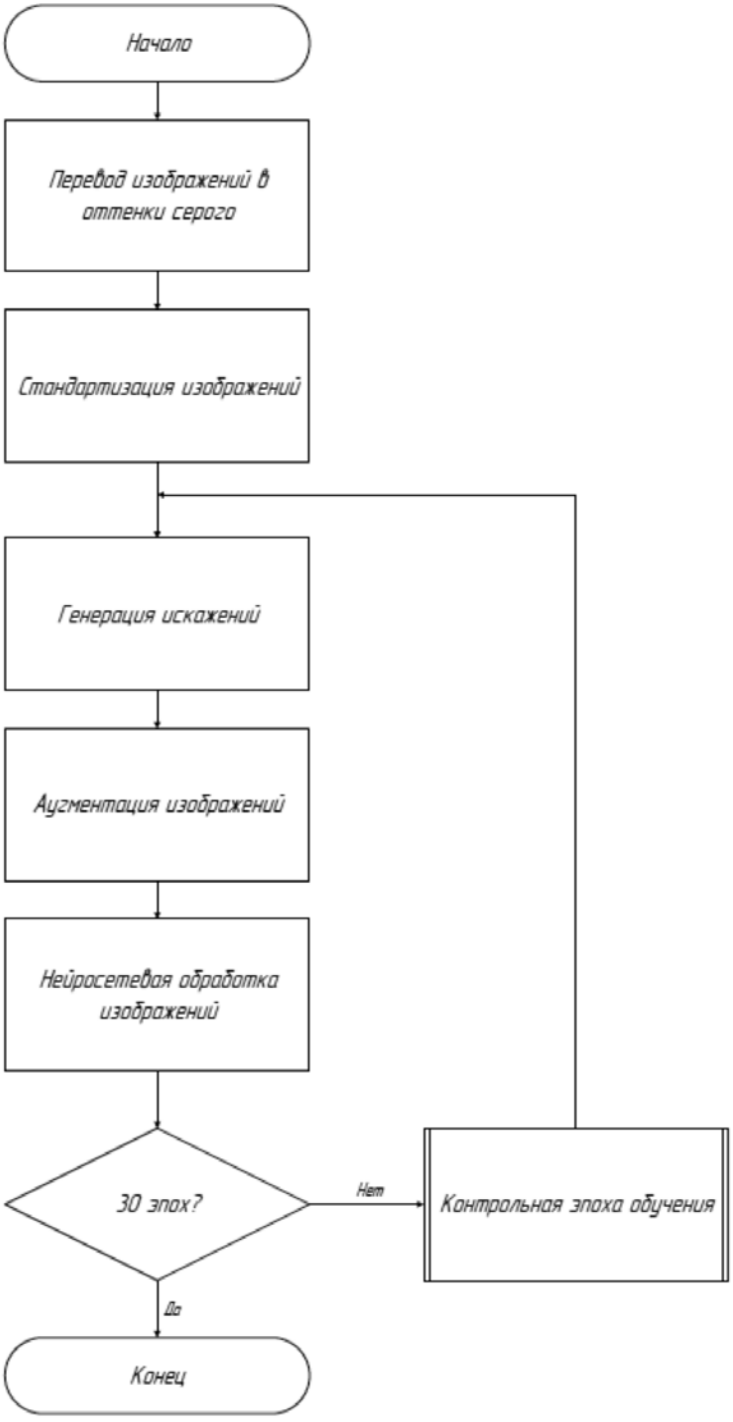

Рис. 2. Обучение сверточной нейронной сети

Рис. 3. Фрагмент искусственно созданного набора изображений

Аугментация. Переобучение происходит в основном, когда набор обучающих данных слишком мал. Один из способов, позволяющий исправить эту проблему, заключается в расширении набора данных (так называемое увеличение) до необходимого количества обучающих примеров. Улучшение данных – это процесс применения произвольных преобразований к исходным данным и добавления полученных измененных данных в исходный обучающий набор. Цель состоит в том, чтобы сгенерировать реалистичные изображения, которые модель никогда не увидит в последующем: ни в наборе данных для валидации, ни в тестовом.

В tf.keras расширение данных осуществляется с помощью класса ImageDataGenerator. Достаточно передать конструктору класса набор различных значений для необходимых параметров:

– width_shift_range = 0,1 – задает случайный сдвиг по ширине (доля от ширины изображения, если < 1, или количество пикселей, если >= 1);

– height_shift_range = 0,1 – задает случайный сдвиг по высоте (доля от высоты изображения, если < 1 или количество пикселей, если >= 1);

– zoom_range = 0,2 – Float или [lower, upper]. Диапазон случайного выбора масштабирования изображения (если float, то [lower, upper] = [1-zoom_range, 1+zoom_range]);

– shear_range = 0,1 – Float, задает диапазон сдвига пикселей изображения (угол сдвига в градусах в направлении против часовой стрелки).

Окончательное изображение фиксированного размера получается с использованием билинейной интерполяции искаженного входного изображения (рис. 3) [5]. Эти искажения позволяют обучать нейронную сеть без переобучения и значительно улучшить производительность обобщения результатов (то есть, как показывает практика, частота ошибок на первом этапе обучения уменьшается с 3,98 до 0,71).

Параметры при обучении сети. Обучение нейронной сети осуществляется с помощью метода fit, основными аргументами которого являются [6]:

– X_train – массив обучающих данных (модель имеет один входной слой).

– Y_train – массив меток (модель имеет один выходной слой).

– Batch_size (размер пакета обучения) – градиент усредняется по примерам, имеющимся в обучающем пакете, после чего пересчитываются веса нейронной сети с использованием этого усредненного значения. В определенном смысле градиент аппроксимируется по всему тренировочному набору (batch_size = 50).

– Steps_per_epoch – количество шагов (часть данных), после выполнения которых эпоха обучения считается завершенной (Steps_per_epoch = 2000).

– Epochs – число итераций (эпох) применения всех заданных параметрами x и y данных.

– Validation_data – кортеж (x_validation, y_validation), который используется для оценки потерь по завершению каждой эпохи обучения.

– Shuffle – параметр типа Boolean, равный 1 – флаг перетасовки обучающих данных перед началом каждой эпохи обучения.

Для обучения и оценки модели исходный набор данных, состоящий из 42689 изображений, был разделен на обучающий (Train) и тестовый (Test) наборы данных в соотношении 80:20 соответственно. Обучающий набор, в свою очередь, включает в себя проверочный набор данных (Validation), состоящий из 6831 изображения. Таким образом, исходный набор данных подразделяется на: *обучающий набор* (Trainig) – *27320 изображений*; *проверочный набор* (Validation) – *6831 изображение*; *тестовый набор* (Test) – *8538 изображений*.

На этапе обучения сеть обрабатывает пакет из 2000 изображений из набора данных train за одну итерацию (эпоху). После успешного обучения точность вычисляется с использованием всех изображений из тестового набора данных. На рис. 4 показано, что точность классификации растет с увеличением числа обучающих итераций. График показывает, что, начиная со второй итерации, сеть достигает точности классификации выше 0,9.

Результаты обучения. Для реализации и обучения модели использовались библиотеки глубокого обучения TensorFlow и Keras совместно с библиотеками Cuda и Cudnn для ускоренного обучения сети на GPU.

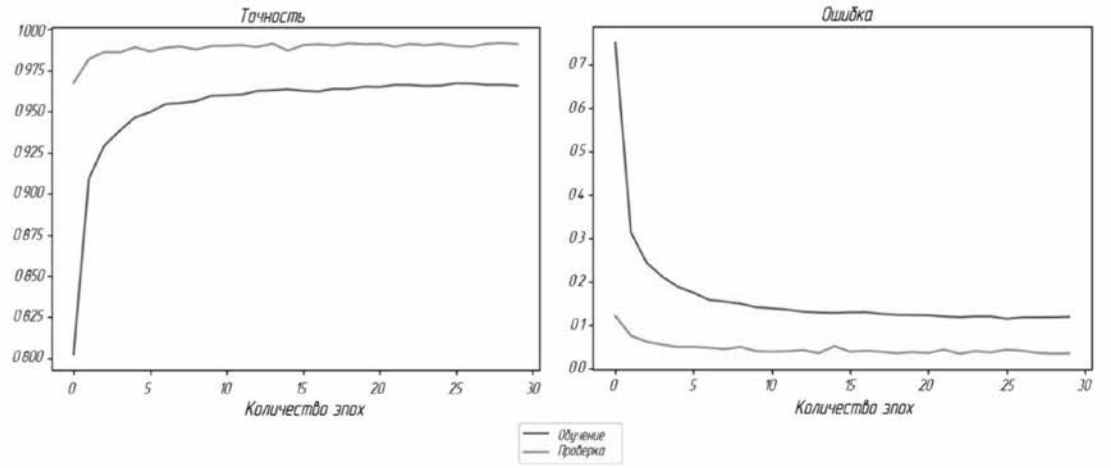

Рис. 4. Показатели точности и потерь на протяжении 30 эпох обучения

Модель обучалась с использованием системы на базе процессора AMD Ryzen 5 2600 Six-Core Processor 3.4 GHz, 16 GB оперативной памяти класса DDR4 3200 MHz и графического процессора NVIDIA GeForce GTX 1660 6GB GDDR5 с частотой ядра/памяти: 1830/8000 MHz соответственно. Обучение модели на GPU заняло 15,5 мин, а время обучения на CPU составило 4,2 ч.

В результате обучения нейронной сети на протяжении 30 эпох были получены следующие данные: точность на обучающих данных – 0,9623; потери (ошибка) на обучающих данных – 0,1304; точность на проверочных данных – 0,9918; потери (ошибка) на проверочных данных – 0,0319. Для наглядности с помощью библиотеки Matplotlib были построены графики точности и потерь (рис. 4).

Реализация модели достигает точности 99,22 %, что превышает результат из исследования [7] Rinu Gour, где сверточная нейронная сеть, обученная на базе дорожных знаков Германии GTSRB, достигает точности 95 %. Модель сверточной сети в исследовании Rinu Gour, построенная для распознавания 43 классов дорожных знаков Германии, состоит из сверточных слоев и слоев подвыборки. На каждом слое из изображения извлекаются признаки (цвет, форма и др.), которые помогают классифицировать изображение. Также в модели предусмотрен слой отсева, который используется для предотвращения переобучения нейронной сети. В результате 15 эпох обучения были получены высокие показатели точности на обучающих и проверочных данных: 0,948 и 0,98 соответственно.

Заключение

В результате обучения нейронной сети, которое длилось на протяжении 30 эпох, была достигнута точность предсказания в 99,22 %, что превышает результат из исследования Rinu Gour, где нейронная сеть достигает точности 95 %. Следовательно, можно сделать вывод о том, что использования простой модели глубокой классификации недостаточно для повышения точности предсказания. Перед обучением очень важны всевозможные виды предварительной обработки изображений. В предлагаемой модели используется предварительная обработка размытых и зашумленных изображений, а также аугментация данных, что, в свою очередь, позволяет улучшить точность предсказания.

Список литературы

1. Смольянинов В.А., Гришунов С.С., Белов Ю.С. Подходы к проектированию и разработке программного детектора дорожных знаков // Сборник избранных статей по материалам научных конференций ГНИИ «Нацразвитие»: материалы Международных научных конференций. 2020. С. 145–149.

2. Смольянинов В.А., Белов Ю.С. Сравнительный анализ алгоритмов поиска дорожных знаков // Высокие технологии и инновации в науке: сборник избранных статей Международной научной конференции. 2020. С. 186–190.

3. Шахуро В., Конушин А. Российская база изображений автодорожных знаков // Компьютерная оптика. 2016. № 2. С. 294–300.

4. Yasmina D., Karima R., Ouahiba A. Traffic signs recognition with deep learning. 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, 2018. P. 1–5.

5. Shawon A., Jamil-Ur Rahman M., Mahmud F., Arefin Zaman M.M. Bangla handwritten digit recognition using deep cnn for large andunbiased dataset. Proceedings of 2018 International Conference on Bangla Speechand Language Processing (ICBSLP), Sep. 2018. P. 1–6.

6. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и Tensorflow / Пер. с англ. Слинкин А.А. М.: ДМК Пресс, 2018. 294 с.

7. Class Data Science Project 2020 – Traffic Signs Recognition [Electronic resource]. URL: https://medium.com/dataflair/class-data-science-project-for-2020-traffic-signs-recognition-12b09c131742 (date of access: 11.05.2021).