

Система распознавания расширенного набора знаков дорожного движения

Е. М. Мачеев, А. В. Девяткин*, А. Р. Музалевский

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

*avdevyatkin@etu.ru

Аннотация. Решение задачи определения и распознавания знаков дорожного движения является одной из важнейших тем исследований для создания беспилотного транспорта. В работе описывается общая реализация подсистемы определения знаков и предлагается решение проблемы отсутствия отдельных знаков дорожного движения в открытых датасетах, основанное на использовании свёрточных нейросетей.

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

I. ВВЕДЕНИЕ

Увеличение количества транспортных средств, управляемых человеком, ведет к, соответственно, увеличению количества аварий в абсолютных значениях. В современном мире можно наблюдать рост интереса к автономным транспортным средствам. Несмотря на то, что на данный момент на рынке не представлены полностью автономные автомобили, подобную тенденцию нельзя не замечать. На этом промежуточном этапе разрабатываются, патентуются и тестируются различные модули подобных автономных систем. Некоторые из них активно применяются в качестве вспомогательных систем. Наиболее широкой и тесно взаимодействующей с водителем транспортного средства (ТС) является, пожалуй, ADAS (полн. advanced driver-assistance systems) – усовершенствованная система помощи водителю, которая может включать следующие элементы:

- адаптивный круиз контроль;
- адаптивное управление освещением;
- автоматическая система торможения (АБС);
- автоматическая парковка;
- мониторинг слепых зон;
- мониторинг пешеходов;
- мониторинг расстояния;
- обнаружения скольжения автомобиля;
- обнаружения сонливости водителя;
- система помощи избегания столкновения;
- помощь в следовании в соответствии с указанием разметки;
- предупреждение о выезде с полосы движения;
- система распознавания знаков/препятствий.

Приведенный список не полный. Производители могут как включать, так и не включать какие-либо элементы из перечисленных в своей системе.

Безусловно, до определенного периода, необходимым компонентом автономным транспортным средствам будет являться как раз последний пункт – система распознавания дорожных знаков. Этой теме – вопросу о проектировании подобных систем, как и в целом, анализу безопасности движения, посвящено множества статей, например [2, 3]. Обе статьи посвящены описанию подходов и алгоритмов распознавания.

На данный момент существует множество датасетов знаков, позволяющие обучать классификаторы. Так, в общем доступе можно найти:

1. Russian traffic sign images dataset (RTSD) – датасет российских знаков. По разработке датасета написана статья [4]. Стоит отметить, что работа, проделанная при формировании этого датасета колоссальна, итоговый датасет занимает более 18 Гб (включая данные для детектора), что предоставляет значительно больше данных по сравнению с частью зарубежных разработок, о которых пойдет речь ниже. Стоит заметить, что несмотря на колоссальный размер, датасет не охватывает все существующие знаки, а ограничен 106 классами, в которые не попали, например:
 - Знак «1.6» – пересечение равнозначных дорог;
 - Знак «1.31» – туннель;
 - Знак «2.4» – предписание «уступите дорогу»;
 - Знак «3.21» – конец запрещения обгона;
 - Знак «3.22» – обгон грузовым автомобилям запрещен;
 - Знак «3.23» – конец запрещения обгона грузовым автомобилям;
 - Знак «3.24» (90, 100, 110, 120, 130) – ограничение максимальной скорости;
 - Знак «3.25» – конец ограничения максимальной скорости;
 - Знак «3.31» – конец всех ограничений;
 - Знак «6.3.2» – зона для разворота.
2. German Traffic Sign Recognition Benchmark (GTSRB) – датасет немецких знаков. Несмотря на отличия в некоторых реализациях знаков (например, толщина окантовочной линии для знаков ограничения максимальной скорости, размер и форма цифр), данный датасет так же представляет интерес при разработке системы. Полностью отличающиеся знаки можно исключить из итоговой комбинации датасетов при обучении

классификатора, получив тем самым больше различных данных. Но, даже комбинация с RTDS не покрывает вышеперечисленные отсутствующие знаки.

Стоит отметить также, что существуют и другие наборы данных по тематике: BelgiumTS Dataset [5], Chinese Traffic Sign Database [6]. Однако проблема нехватки данных остается.

В работе рассматривается задача разработки классификатора, способного работать с классами, данных для которых нет в общем доступе. Задача актуальна, поскольку сборка собственного датасета как правило трудоемкий и кропотливый процесс, занимающий немало времени. Первоначальную задачу, стоящую перед авторами статьи можно сформулировать так: разработать детектор и классификатор для определенного перечня знаков. Проблемой, с которой столкнулись авторы, является недостаточный набор данных для разработки классификатора. Было предложено разработать некоторый энкодер знаков, то есть систему, преобразующую входное изображение в некоторый вектор признаков. Выходной вектор признаков уникален (то есть в некоторой степени отличается) для каждого класса знаков. Предполагается, что можно разработать систему, которая будет классифицировать, выделять и объединять в классы знаки, которые не участвовали в обучении.

II. ОПИСАНИЕ МОДЕЛИ

Опустив подзадачу детекции, как элемент изначальной задачи, стоящий перед авторами, будем охватывать лишь модель энкодера, на базе которого будет строиться классификатор.

Энкодер – функция (или система), получающая на вход исходные данные и преобразующая их в некоторые данные, как правило, меньшей размерности. Декодер – функция (или система), выполняющая обратное преобразование, то есть из меньших данных, восстанавливающая исходные. Наиболее прикладной задачей для этих двух систем является сжатие данных. На рисунке 1 представлена схема их взаимодействия.

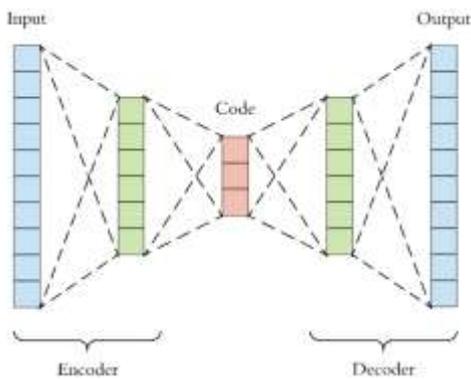


Рис. 1. Схема Энкодер-Декодер

При этом, процесс кодирования-декодирования может совершаться как с потерями данных (например, сжатие мультимедийных данных с переходом на меньший битрейт, частоту квантования или снижение глубины цвета), так и без потерь (различные алгоритмы LZW, LZMA, ALAC, точно ставящие в соответствие некоторый код исходным данным).

С этой точки зрения, можно относиться к закодированным данным как к некоторой промежуточной величине, несущей достаточно данных для восстановления, в некоторой степени, исходных данных, однако имеющих существенно меньшую размерность. Также, удобно воспринимать выходное значение энкодера как некоторую точку в n -мерном пространстве. Например, для энкодера, используемого при решении описываемой задачи, выходным является вектор размерности 512. То есть, любому входному изображению ставится в соответствие эта точка в 512-мерном пространстве.

Для реализации энкодера используется библиотека PyTorch Metric Learning [7]. Функцией потерь при обучении является TripletMarginLoss, стремящаяся минимизировать функционал вида:

$$L_{triplet} = \max\{d_{ap} - d_{an} + m, 0\} \quad (1)$$

где d_{ap} , d_{an} – евклидовы расстояния между некоторым якорным значением класса A и позитивным и негативным входным значением;

$$d_{ap} = \|X_a - X_p\|^2, d_{an} = \|X_a - X_n\|^2 \quad (2)$$

m – некоторая положительная величина, характеризующая расстояние между позитивным и негативным примером. Необходима для формирования триплетов из набора данных.

Визуализация представлена в правой части рис. 2.

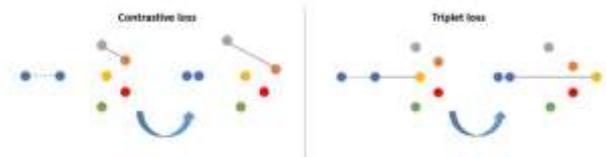


Рис. 2. Схема различных функций потерь

Использование указанной функции потерь позволяет при относительно малом потреблении памяти для расчета (так как при вычислении используются только 3 примера из выборки: якорное значение, позитивный и негативный примеры), эффективно кластеризовать входные изображения. Цель TripletLoss-функции – минимизация расстояния от якоря до позитивного примера и максимизация расстояния от якорного значения до негативного примера. Для сравнения, на рис. 2 представлена другая функция потерь – Contrastive loss. Она также использует попарную разницу между якорным (см. формулу 3, величина d – аналогично по смыслу той, что введена для формулы 1), позитивным и негативным примером, однако не стремится максимизировать расстояние между позитивным и негативным примером;

$$L_{contrastive} = y d_{x_1 x_2}^2 + (1 - y) * \max\{0, m - d_{x_1 x_2}\}^2 \quad (3)$$

В результате обучения только на датасете RTDS, энкодер кодирует входное изображение в точку в 512-мерном пространстве. При этом изображения одного класса находятся в некоторой области этого пространства. Для наглядности, на рис. 3 представлена визуализация кластеров в трехмерном пространстве.

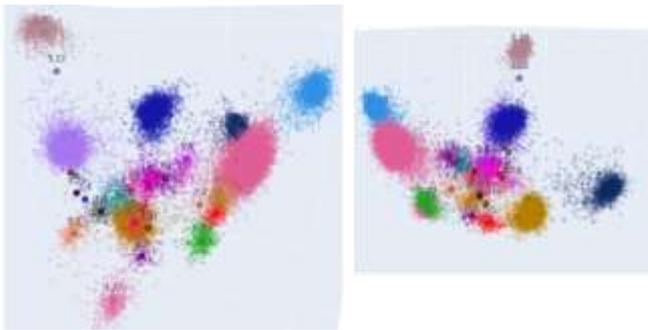


Рис. 3. Визуализация кластеров в трехмерном пространстве

Для проецирования 512-мерного пространства в трёхмерное используется метод главных компонент (Principal Component Analysis – PCA). При этом, конечно, теряется часть информации о расстояниях.

III. ОПРЕДЕЛЕНИЕ ЗНАКА

Как можно видеть на рис. 3, энкодер позволяет кластеризовать знаки. При этом, повторимся, он обучен на большом количестве знаков из датасета RTDS, который охватывает не все требуемые знаки. Подадим ему на вход произвольный знак, класс которого отсутствует в обучающей выборке. Он должен отобразиться некоторой точкой в выходном пространстве, которая, возможно, будет расположена недалеко (по евклидову расстоянию), от некоторых похожих знаков, участвовавших в обучении. Результат представлен на рис. 4.

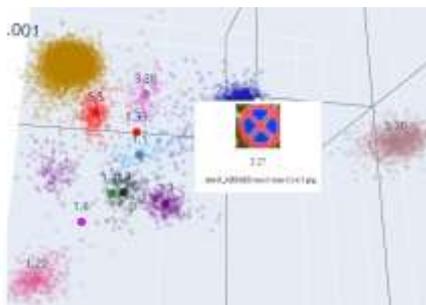


Рис. 4. Визуализация кластеров в трехмерном пространстве

Знак 1.31 и знак 6.3.2 отсутствуют в обучающей выборке, но, как можно видеть, энкодер отображает их в координаты центроидов – среднего арифметического координат в 512-мерном пространстве для каждого класса в обучающей выборке. Для знаков, отсутствующих в обучающей выборке, возьмем некоторое опорное значение, в виде выхода энкодера по образцовому примеру нового знака. Например, для знаков 6.3.2 и 1.31 опорными значениями будут следующие изображения (рис. 5).



Рис. 5. Опорные изображения знаков 6.3.2 и 1.31

По итогу, мы имеем возможность добавлять произвольный знак, который отображается энкодером в некоторую область 512-мерного пространства. Также, для каждого знака мы имеем центроид. Для наглядности, визуализация центроидов представлена на рис. 6.



Рис. 6. Опорные изображения знаков 6.3.2 и 1.31

Повторимся, что для получения центроидов знаков используется арифметическое среднее выходных величин значений выхода энкодера для обучающей выборки, для получения центроидов для данных, отсутствующих в обучающей выборке достаточно подать на вход энкодеру единственное опорное значения нового знака. Для каждого нового входного изображения будем рассчитывать расстояния до всех центроидов, и определять принадлежность знака по наиболее близко расположенному центроиду.

Оценим работоспособность метода в рамках этой задачи. В таблицах ниже представлены метрики классификатора, основанном на энкодере и применении описанного выше метода – классификация на основе ближайшего соседа-центроида.

ТАБЛИЦА I МЕТРИКИ ДЛЯ КЛАССОВ, ПРЕДСТАВЛЕННЫХ В ОБУЧАЮЩЕЙ ВЫБОРКЕ

	Precision	Recall	F1	Количество
1.1	0,994	0,976	0,985	167
1.22	0,996	1	0,998	756
1.31	0,5	0,5	0,5	4
1.8	0,989	0,986	0,988	283
2.1	0,999	0,995	0,997	8963
2.2	0,857	0,991	0,919	217
2.3	0,983	1	0,991	346
2.5	0,991	1	0,995	432
3.1	0,986	0,996	0,991	792
3.18	0,986	1	0,993	218
3.20	0,998	0,998	0,998	1629
3.21	0,4	0,8	0,533	5
3.23	0,2	0,333	0,25	3
3.24	0,999	0,993	0,996	6190
3.27	0,999	0,994	0,996	5127
3.28	0,931	0,996	0,963	272
4.1.1	0,996	0,996	0,996	2688
4.3	0,94	1	0,969	140
5.16	0,998	0,996	0,997	3173
5.19.1	0,999	0,996	0,997	22138
5.20	0,922	0,995	0,957	803
5.5	0,988	0,988	0,988	404
5.6	0,991	0,995	0,993	213
6.4	0,964	0,991	0,977	430
7.3	0,998	0,993	0,995	440
7.4	1	1	1	58

ТАБЛИЦА II МЕТРИКИ ДЛЯ КЛАССОВ, НЕ ПРЕДСТАВЛЕННЫХ В ОБУЧАЮЩЕЙ ВЫБОРКЕ

	Precision	Recall	F1	Количество
1.31	0,5	0,5	0,5	4
1.6	0,5	0,667	0,571	3
2.4	0,833	1	0,909	5
3.22	0,063	1	0,118	1
3.25	0,4	0,75	0,522	8
3.31	0,857	0,667	0,75	9
6.3.2	0,538	0,778	0,636	9

Матрица ошибок представлена на рисунке ниже:

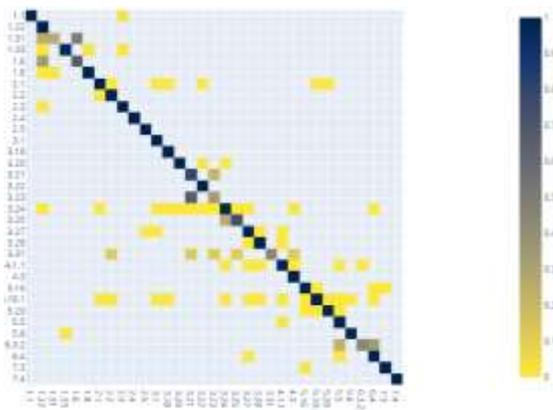


Рис. 7. Матрица ошибок определения знаков

Проведем анализ ошибок:

1.31 – классификатор может с большой вероятностью определить знак как 1.22 или 1.6. Все знаки треугольные с изображением по центру.

1.6 – классификатор может с большой вероятностью определить знак как 1.22. Оба знака треугольные с изображением по центру.

2.4 – хорошо классифицируется описанным методом.

3.22 – плохо классифицируется описанным методом. Часто определяется как 3.20.

3.25 – местами пересекается с 3.24, то есть со знаком окончания ограничения максимальной скорости.

3.31 – плохо классифицируется описанным методом. Часто определяется как 3.21, 3.23, 3.25. Все перечисленные знаки круглые, диагонально перечеркнутые.

6.3.2 – зона для разворота может быть ошибочно воспринята как другие большие синие знаки: парковка и дорога с односторонним движением.

Датасет RTDS выделяет знаки ограничения скорости в отдельные классы для разных скоростей (например, 3.24.10, 3.24.40, 3.24.60, и т. д.). Попробуем переобучить энкодер, добавив их. Такое переобучение позволит энкодеру выделять больше количество признаков в целом, что может благоприятно сказаться на точности дополнительных знаков. Метрики по дополнительным знакам после переобучения представлены в табл. 3.

ТАБЛИЦА 3 МЕТРИКИ ДЛЯ КЛАССОВ, НЕ ПРЕДСТАВЛЕННЫХ В ОБУЧАЮЩЕЙ ВЫБОРКЕ

	Precision	Recall	F1	Количество	Качество
1.31	1,00	0,50	0,67	4	Выросло
1.6	1,00	0,67	0,80	3	Выросло
2.4	1,00	1,00	1,00	5	Выросло
3.22	0,20	1,00	0,33	1	Выросло
3.25	0,45	0,63	0,53	8	Выросло
3.31	0,50	0,89	0,64	9	Уменьшилось
6.3.2	0,67	0,22	0,33	9	Уменьшилось

Как можно видеть, при обучении с большим количеством классов, классификатор на базе энкодера дает более точный результат для некоторых знаков. Причины ошибок остаются прежними. Для решения задачи классификации знаков 3.22, 3.31, 6.3.2, необходимо применить отдельный паттерн-матчинг, в особенности для 5.5, 6.3.2, 6.4 (рис. 8).



Рис. 8. Изображения знаков 5.5, 6.3.2, 6.4 соответственно

Так как знаки похожи, но в обучении 6.3.2 не участвовал, а набора 5.5 и 6.4, видимо, не хватило для того, чтобы энкодер выделял специфичные признаки, которых хватило бы для классификации 6.3.2.

IV. ЗАКЛЮЧЕНИЕ

Предложенная схема показывает работоспособность. Показатели качества приведены в табл. 1, 2 и 3. Совершенно точно можно сказать, что подход дает свои результаты, несмотря на то, что местами точность относительно не высока.

В результате работы получен классификатор на основе энкодера, способный без переобучения классифицировать дорожные знаки, не участвовавшие в обучении.

В дальнейшем планируется использование OCR (optical character recognition) для более детальной классификации знаков начала ограничения скорости и конца ограничения скорости. Возможно, подобным образом удастся повысить точность классификации знака 6.3.2, так как текст на этом знаке находится всегда и в заранее известном месте.

Альтернативным подходом будет переобучение, с добавлением в обучающую выборку дополнительных знаков (так, например, при обучении энкодера использовался не весь датасет, а только знаки, перечисленные в табл. 1). Это позволит энкодеру более эффективно извлекать некоторые признаки, которые в целом улучшат его качество.

СПИСОК ЛИТЕРАТУРЫ

- [1] YOLO: Real-Time Object Detection, URL: <https://pjreddie.com/darknet/yolo/> (Access date: 5.11.2021)
- [2] Гильманов Т.А. «Система распознавания дорожных знаков», КНИТУ-КАИ им. А.Н. Туполева, [3] URL: <https://cyberleninka.ru/article/n/sistema-raspoznaniya-dorozhnyh-znakov-1/viewer> (Access date: 03.03.2022)
- [4] Шемарулин И.А., Карпычев В.Ю. «Распознавание дорожных знаков на основе цветных контуров» (Нижегородский государственный технический университет им. П.Е. Алексеева), URL: <https://cyberleninka.ru/article/n/raspoznavanie-dorozhnyh-znakov-na-osnove-tsvetnyh-konturov/viewer> (Access date: 03.03.2022)
- [5] Russian traffic sign images dataset. V.I. Shakhuro, A.S. Konushin. (NRU Higher School of Economics, Lomonosov Moscow State University), Moscow, Russia. URL: <http://computeroptics.ru/eng/KO/PDF/KO41-ENG%20-17/400221.pdf> (Access date: 03.03.2022)
- [6] BelgiumTS Dataset. Radu Timofte. URL: <https://btsd.ethz.ch/shareddata/> (Access date: 09.11.2021).
- [7] Chinese Traffic Sign Database. National Nature Science Foundation of China(NSFC). URL: <http://www.nlpr.ia.ac.cn/pal/trafficedata/recognition.html> (Access date: 09.11.2021)
- [8] Библиотека PyTorch Metric Learning. URL: <https://kevinmusgrave.github.io/pytorch-metric-learning/> (Access data 03.03.2022).
- [9] Связь дисперсии по компонентам и потери информации при применении метода главных компонент. URL: <https://stats.stackexchange.com/questions/271979/does-applying-pca-result-in-loss-of-information> (Access data 03.03.2022).