УДК 004.855.5
doi:10.31799/1684-8853-2021-4-61-70

Научные обзоры
Surveys

Формирование обучающей выборки в задачах машинного обучения. Обзор

А. В. Парасич[а], *инженер-программист, orcid.org/0000-0003-2728-0893 parasichav@yandex.ru*
В. А. Парасич[б], *канд. техн. наук, доцент, orcid.org/0000-0003-3593-2345*
И. В. Парасич[б], *канд. техн. наук, доцент, orcid.org/0000-0003-1965-8737*
[а]*ООО «ТРИДИВИ», Ленина пр., 64, Челябинск, 454080, РФ*
[б]*Южно-Уральский государственный университет, Ленина пр., 76, Челябинск, 454080, РФ*

Введение: *правильное формирование обучающей выборки является ключевым фактором при решении задач машинного обучения. При этом в реальных обучающих выборках часто встречаются те или иные трудности и ошибки формирования выборки, оказывающие критическое влияние на результат обучения. Проблема формирования обучающей выборки возникает во всех задачах машинного обучения, поэтому знание возможных вопросов при формировании обучающей выборки будет полезно при решении любой задачи машинного обучения.* ***Цель:*** *обзор возможных проблем формирования обучающей выборки с целью облегчить их обнаружение и устранение при работе с реальными обучающими выборками. Анализ влияния этих проблем на результат обучения.* ***Результаты:*** *проведен обзор возможных ошибок формирования обучающей выборки, таких как отсутствие данных, разбалансировка, ложные внутривыборочные закономерности, формирование выборки из ограниченного набора источников, изменение генеральной совокупности во времени и др. Исследовано влияние этих ошибок на результат обучения, а также на формирование тестовой выборки и измерение качества алгоритма обучения. Pseudo-labeling, data augmentation, hard samples mining рассматриваются как наиболее эффективные способы расширения обучающей выборки. Предложены практические рекомендации по формированию обучающей и тестовой выборок. Приведены примеры из практики соревнований Kaggle. Рассмотрена проблема cross-dataset generalization. Предложен алгоритм решения проблемы cross-dataset generalization при обучении нейронных сетей, названный Cross-Dataset Machine, простой в реализации и позволяющий получить выигрыш в cross-dataset обобщении.* ***Практическая значимость:*** *материалы статьи могут использоваться в качестве практического руководства при решении задач машинного обучения.*

Ключевые слова *— машинное обучение, обучающая выборка, Kaggle, глубокие нейронные сети, деревья решений, ImageNet.*

Для цитирования: Парасич А. В., Парасич В. А., Парасич И. В. Формирование обучающей выборки в задачах машинного обучения. Обзор. *Информационно-управляющие системы*, 2021, № 4, с. 61–70. doi:10.31799/1684-8853-2021-4-61-70
For citation: Parasich A. V., Parasich V. A., Parasich I. V. Training set formation in machine learning tasks. Survey. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2021, no. 4, pp. 61–70 (In Russian). doi:10.31799/1684-8853-2021-4-61-70

Введение

Принципиальным недостатком большинства теоретических построений из области машинного обучения и статистики [1] (в том числе рассуждений о переобучении и измерении качества работы алгоритма) является заложенное в них предположение о том, что обучающая выборка представляет идеальную независимую репрезентативную выборку из генеральной совокупности, отражающую все ее свойства. В действительности подавляющее большинство реальных обучающих выборок содержат те или иные погрешности в формировании выборки, начиная от отсутствия данных определенного вида, заканчивая ложными закономерностями внутри данной выборки. И эти проблемы формирования выборки часто оказывают решающее влияние на качество алгоритма. При этом с точки зрения успеха в решении задач машинного обучения качество данных, как правило, намного важнее качества алгоритма обучения. Именно устранение проблем формирования обучающей выборки часто является ключом к успешному решению задачи распознавания. Рассмотрим наиболее распространенные проблемы и ошибки при формировании обучающей выборки [2, 3].

Проблемы формирования обучающей выборки

Отсутствие данных определенного типа. Самая простая, самая распространенная и самая опасная проблема. Если данных определенного типа (или из определенного участка пространства признаков) в обучающей выборке $X^L = (x_i, y_i)_{i \in 1..L}$ нет (например, в задаче распознавания лиц в выборке нет людей в очках), то алгоритм не сможет научиться правильно работать на этих данных. Как правило, очень трудно гарантировать, что в выборке есть данные всех возможных типов, поэтому с проблемой отсутствия данных приходится постоянно бороться. Следовательно, одной из главных ценностей при решении задач машинного обучения является изучение обучающей выборки.

Слишком мало данных определенного типа. Если данных определенного типа в обучающей выборке $X^L = (x_i, y_i)_{i \in 1..L}$ слишком мало (например, один объект), то крайне высока вероятность, что алгоритм не сможет выучить по этим данным правильные закономерности $y = f_k(x_i)$, а вместо этого выучит первую попавшуюся закономерность (наподобие того, что если левый верхний пиксель изображения — красный, то это изображение собаки, иначе — изображение кошки). При малом числе данных в обучающей выборке для их классификации внутри обучающей выборки подойдет практически любое (первое попавшееся) решающее правило. Поэтому данных в выборке должно быть достаточно в той мере, чтобы ошибочные закономерности случайно не позволили правильно классифицировать эти данные внутри обучающей выборки.

Разбалансировка. Нарушение равномерного количественного соотношения данных разных типов. Простейший пример — разное количество обучающих примеров разных классов $C_1 ... C_n$ в задаче классификации [4] (например, в задаче распознавания пола человека по изображению лица в выборке мужчин гораздо больше, чем женщин, тогда алгоритму может быть выгоднее всегда говорить, что на изображении мужчина). Более сложный пример — разбалансировка по некоторой оси разнообразия данных (в задаче распознавания пола человека по изображению лица в выборке гораздо больше дневных снимков, чем ночных, из-за чего может пострадать качество распознавания ночных снимков).

Разбалансировка возникает очень часто, поскольку трудно гарантировать соблюдение равномерных пропорций для всех возможных осей вариативности данных. При этом может очень негативно сказаться на качестве обучения, приводя к необоснованному с семантической точки зрения завышению влияния одних примеров и занижению влияния или полному игнорированию других примеров. Так происходит потому, что в алгоритмах обучения в процессе обучения оптимизируются метрики качества, представляющие собой простую сумму ошибки по всем обучающим примерам, и эта сумма, понятное дело, будет зависеть от соотношения количеств обучающих примеров разных типов. И те примеры, которых в выборке больше, будут больше влиять на эту ошибку, а значит, и на результат обучения. В предельном случае при резкой разбалансировке слабо представленный класс (тип примера) будет полностью проигнорирован и никогда не будет распознаваться (этот класс никогда не будет возвращаться алгоритмом в качестве ответа). Иногда для этого достаточно 10-кратной разбалансировки.

Проблема разбалансировки решается простым перевзвешиванием обучающих примеров (повышением веса слабо представленным примерам либо понижением веса сильно представленным примерам). Если интерфейс алгоритма обучения не поддерживает веса примеров, то недостаточно представленные примеры можно просто продублировать нужное число раз. Если доступно добавление новых данных, то недостающих данных можно добавить. Сложность заключается в том, что разбалансировку не всегда удается обнаружить. На практике встречаются весьма сложные и нестандартные виды разбалансировок, которые оказывают большое влияние на качество алгоритмов.

Если число примеров разных классов сильно несбалансированно, и при этом известно, что при реальном использовании модели распределение числа примеров разных классов будет таким же несбалансированным, то возникает дилемма. Если сбалансировать выборку, то у алгоритма будет неверное представление о вероятности появления примеров разных классов, из-за чего распределение ответов алгоритма будет смещено относительно истинного распределения классов, что может привести к существенному увеличению числа ошибок. Если же не сбалансировать выборку, то алгоритм может просто проигнорировать слабо представленные классы и не научиться их распознавать. Для обучения глубоких нейронных сетей [5] существует следующий прием: на первых эпохах обучения баланс классов выравнивают, на последних эпохах обучения возвращают тот баланс, который был изначально. Это позволяет улучшить сходимость сети (за счет того, что слабо представленные классы не игнорируются), и в то же время у сети не возникает искаженного представления о вероятности появления примеров разных классов. Также при обучении нейронных сетей для решения проблемы разбалансировки можно использовать *focal loss* [6]. Классическими методами борьбы с разбалансировкой являются *SMOTE* [7] и *ADASYN* [8]. Проблема разбалансировки подробно исследуется в работах [9–13].

Ложные внутривыборочные закономерности. Ложные зависимости, которые существуют только внутри данной обучающей выборки $X^L = (x_i, y_i)_{i \in 1..L}$ и являются следствием неправильного формирования выборки, из-за чего алгоритм может выучить эти ложные зависимости вместо реальных зависимостей $y = f_k(x_i)$ (если эти ложные зависимости будет проще выучить, чем реальные зависимости) и будет некорректно работать за пределами обучающей выборки. Например, в задаче распознавания пола человека по изображению лица в выборке все изображения мужчин сняты в дневное время, а все изображения женщин — в ночное. Тогда алгоритм может научиться определять пол человека исключи-

тельно на основе уровня освещенности изображения и не будет работать в реальных условиях.

Ложные внутривыборочные закономерности могут проявляться в виде ложной внутривыборочной разбалансировки, т. е. жесткой зависимости (все изображения мужчин сняты только в дневное время, а все изображения женщин — только в ночное) внутри обучающей выборки нет, но есть сильный количественный перекос в сторону такой зависимости, из-за чего алгоритм выучит данную ложную закономерность.

Ложные внутривыборочные закономерности второго порядка. Внешняя зависимость, которая не является ложной и реально существует в данных, но не является прямым отражением природы данных, из-за чего качество работы алгоритма в нестандартных условиях может пострадать при заучивании такой зависимости. Например, в задаче детекции пешеходов [14] в абсолютном большинстве случаев пешеход идет по земле. Эта закономерность не является ложной и реально существует в данных. Но если алгоритм выучит, что пешеход всегда идет по земле, то он не будет детектировать прыгающих пешеходов. Поэтому лучше учить те закономерности, которые являются прямым следствием природы данных.

Формирование выборки из ограниченного набора источников («микроисточников»). Очень часто обучающие выборки формируются не путем независимого выбора объектов из генеральной совокупности, а набираются из ограниченного набора источников данных $S_1 ... S_n$, что создает определенные риски при обучении. Алгоритм может выучить конкретные источники данных вместо истинной зависимости $y = f_k(x_i)$. Например, в задаче распознавания спама, если обучающая выборка писем была собрана из ограниченного набора спамерских и «чистых» ящиков и среди признаков есть признак «имя почтового ящика», то алгоритм может выучить конкретные имена спамерских почтовых ящиков и будет некорректно работать для любых имен спамерских ящиков, не представленных в обучающей выборке.

Поэтому в обучении нельзя использовать признаки, все значения которых принципиально не могут быть представлены в обучающей выборке (фамилии, города, названия почтовых ящиков и т. д.).

Формирование выборки из ограниченного набора источников также может стать причиной появления ложных внутривыборочных закономерностей (например, в задаче распознавания пола человека по изображению лица мужчин снимали хорошей камерой, а женщин — плохой, тогда вместо реальных зависимостей алгоритм может научиться определять пол по качеству изображения). Поэтому при таком способе формирования выборки нужно быть предельно внимательным к возможным негативным последствиям.

Разное распределение значений признаков. В более общем виде не все возможные значения признаков представлены в обучающей выборке. Например, в задаче предсказания поведения покупателя в магазине в обучающей выборке только люди молодого возраста, а в тестовой выборке — только люди пожилого возраста. Частный случай проблемы отсутствия данных определенного вида.

Недозаполненность признакового пространства. Пространство признаков f_1, ..., f_p порождает некоторое разбиение обучающей выборки $X^L = (x_i, y_i)_{i \in 1..L}$ на группы примеров $C_1 ... C_n$ в соответствии с тем, какие значения признаков $f_k(x_i)$ у данных примеров. При этом в зависимости от того, какие признаки используются, в некоторые части множества возможных значений признаков может не попасть ни один обучающий пример либо попасть слишком мало обучающих примеров.

В качестве примера можно привести использование гистограммы ориентированных градиентов [15] в задаче детекции объекта. При использовании слишком большого числа угловых ячеек (например, 100 ячеек) может получиться так, что некоторым ячейкам не будет соответствовать ни один объект из обучающей выборки, поэтому алгоритм не сможет научиться детектировать объекты с такой ориентацией. При использовании меньшего числа ячеек (например, 50 ячеек) подобной проблемы может не возникнуть.

Чем более сложные признаки используются, тем выше вероятность того, что некоторые части признакового пространства останутся недозаполненными примерами из обучающей выборки. Поэтому при переходе к более сложным признакам может потребоваться расширение обучающей выборки, чтобы добавление более сложных признаков приводило к росту качества, а не к переобучению [16].

Сбор одинаковых данных при формировании выборки. Часто встречающаяся на практике ошибка. При формировании обучающей выборки собираются слишком одинаковые данные либо варьируются не все степени свободы данных (например, собраны данные не для всех возможных диапазонов расстояний от объекта до камеры), что приводит к неработоспособности алгоритма в тех условиях, данные для которых не были собраны.

Сбор данных не в тех условиях, в которых будет использоваться система. Другая важная с практической точки зрения ошибка. Сбор данных не в тех условиях, в которых будет использоваться система, может, с одной стороны, привести к появлению в данных тех степеней свободы,

которых не будет при реальном использовании системы (что затруднит обучение), а, с другой стороны, может привести к отсутствию в выборке некоторых разновидностей данных, на которых система должна будет работать, из-за чего алгоритм не сможет обучиться правильно работать на этих данных. Подобная ошибка опасна еще и тем, что может увести всю разработку системы в неверном направлении (разработка не будет сконцентрирована на решении тех проблем, которые наиболее важны при реальном использовании системы, и стратегические решения о выборе направления развития системы могут быть приняты неверно). Поэтому с самого начала разработки любой системы распознавания рекомендуется собирать данные в условиях, максимально приближенных к тем, в которых эта система будет использоваться.

Неправильные значения целевой переменной. Предельный случай проблем формирования выборки, тем не менее встречающийся в реальных выборках. Может возникать в результате ошибок разметки.

Смещение малых подвыборок. Даже если обучающая выборка не содержит в себе ни одной из вышеперечисленных проблем, вроде отсутствия данных или присутствия ложных закономерностей, это не значит, что таких проблем нет в ее малых подвыборках. Малые подвыборки образуются при обучении нижних уровней деревьев решений (так как при обучении деревьев решений при обучении каждой следующей вершины ее множество данных делится между ее левым и правым сыном для дальнейшего обучения). При обучении нейронных сетей малой подвыборкой можно считать мини-батч, либо множество обучающих примеров с высокой ошибкой в конце обучения, сильнее всего влияющих на обучение сети на поздних этапах, либо множество примеров, вызывающих активацию определенного нейрона, либо множество примеров, соответствующих определенному внутреннему состоянию сети. При этом наличие рассмотренных проблем в малой подвыборке гораздо более вероятно, чем во всей выборке, следовательно, высок риск неправильного обучения. Это может приводить к трудно диагностируемым, но опасным формам переобучения.

Изменение генеральной совокупности во времени. Даже если у нас есть идеально правильная обучающая выборка $X^L = (x_i, y_i)_{i \in 1..L}$, не содержащая никаких ошибок и проблем формирования выборки (допустим, мы решаем задачу машинного обучения для крупной интернет-компании, используя для обучения датасет из данных о нескольких миллионах пользователей, где проблемы отсутствия данных какого-либо вида не может быть в принципе), это не гарантирует нам полное отсутствие проблем, связанных с формированием выборки. Генеральная совокупность может меняться во времени (если алгоритм работает в онлайн-режиме). Могут появляться новые разновидности данных (новые источники данных), происходить сезонные колебания или локальные всплески определенного рода активности. В данном случае может помочь правильная схема валидации. Например, рекомендуется делать валидацию по времени (данные из последнего месяца — в тестовую выборку, остальные — в обучающую).

Проблемы формирования тестовой выборки

Те же самые проблемы (отсутствие данных, недостаточное количество данных, разбалансировка, ложные зависимости) могут присутствовать и в тестовой выборке, по которой измеряется качество работы алгоритма, из-за чего оценка качества работы алгоритма может оказаться некорректной.

Стандартная процедура тестирования (разбиение множества данных D на обучающую T и тестовую V выборки случайным образом) и кросс-валидация [17] не позволяют обнаружить проблемы формирования выборки.

Если в исходной выборке D отсутствуют данные некоторого вида, то этих данных не будет ни в обучающей, ни в тестовой выборке, поэтому качество работы на этих данных измерено не будет, и отсутствие данных никак не проявится при тестировании.

Если в исходной выборке D присутствует ложная внутривыборочная закономерность, то она будет существовать и в обучающей, и в тестовой выборках, порожденных из выборки D. И если алгоритм выучит эту ложную внутривыборочную закономерность, то на тестовой выборке он сможет отработать корректно, т. е. проблема никак не проявится. Если в выборке нет примеров определенного типа, то и качество работы на них измерено не будет, и эта проблема тоже никак не проявится при тестировании.

Разбалансировка также оказывает влияние на результат тестирования алгоритма. При разной сложности распознавания данных разных типов может получиться так, что качество работы алгоритма на тестовой выборке будет в первую очередь зависеть от баланса количества сложных данных в выборке по отношению к простым данным, нежели от качества алгоритма.

Если в тестовой выборке присутствуют неправильные значения целевой переменной, то качество алгоритма будет просто неправильно измерено, независимо от предположений о пра-

вильности / неправильности закономерностей. Данный пример наглядно показывает, что не стоит слепо полагаться на результаты тестирования, не анализируя суть происходящего и возникающие ошибки.

Например, если выборка составлена из данных из заранее известного набора источников D_1, D_2, ..., D_n и известно, что в ходе использования системы могут появляться новые источники D_i, то для проверки обобщающей способности алгоритма выборку следует разбивать на обучающую и тестовую по источникам (например, данные из источников D_1, D_2, ..., D_k — в обучение, а данные из источников D_{k+1}, D_{k+2}, ..., D_n — в тест).

Эффективные способы расширения обучающей выборки

Добавление данных в обучающую выборку является, как правило, самым эффективным способом повышения качества обучения. При этом не стоит забывать о том, что лучше всего добавлять именно те данные, которых не хватает в обучающей выборке и с распознаванием которых алгоритм испытывает проблемы. И не создать дополнительную разбалансировку либо ложные внутривыборочные закономерности. Рассмотрим наиболее распространенные способы добавления данных в обучающую выборку.

Pseudo-labeling. Noisy Student. В последнее время в соревновательной практике широко используется *pseudo-labeling* [18] (метод *self-supervised learning* [19], когда алгоритм сначала обучается на коллекции размеченных данных, затем ответы обученного алгоритма используются для разметки набора неразмеченных данных, и далее полученные данные используются для дообучения алгоритма, при этом используются только те данные, в ответах на которые алгоритм уверен). Разновидности pseudo-labelling — алгоритмы *Noisy Student* [20] и *Meta Pseudo Labels* [21] — показали одни из лучших результатов на *ImageNet* [22].

Data augmentation. Важный этап обучения нейронных сетей, состоящий в модификации обучающих изображений (поворот, масштабирование, зеркальное отражение и т. д.) по определенному правилу с целью расширить обучающую выборку и повысить ее разнообразие. Рассмотрим наиболее эффективные виды аугментаций.

Аугментации цвета и контраста. Случайное изменение компонент R, G, B цвета пикселей изображения. Один из самых эффективных методов аугментации данных, потому что нейросети без этой аугментации имеют тенденцию к заучиванию фич вида «сумма цветов пикселей в области».

Аугментации масштаба, random cropping. Практически всегда приводят к улучшению качества. В работе [23] показано повышение качества детекции объектов на 10 % на датасете *COCO* [24] благодаря добавлению аугментаций масштаба. Эффективность данного вида аугментаций объясняется тем, что сверточные нейронные сети по своей природе не инвариантны к масштабу, а изменение масштаба изображения значительно повышает разнообразие данных с точки зрения нейросети.

Еще одним эффективным видом аугментации являются *CutOut* [25] и *Random Erasing* [26] (закрашивание случайных прямоугольников на картинке, чтобы нейросеть не могла научиться распознавать объект по одной конкретной детали внешнего вида, например, распознавать машину по колесу).

Одним из эффективных приемов обучения нейросетей является постепенное уменьшение интенсивности аугментаций по ходу обучения. Это позволяет нейросети лучше адаптироваться к исходному распределению, в то же время улучшает сходимость и устойчивость сети за счет повышения разнообразия данных. Также существуют методы автоматического подбора наиболее эффективных аугментаций под заданную выборку [27]. Эффективность различных схем аугментации исследуется в работах [28–30].

Hard Samples Mining. Классическая проблема при обучении детектора объектов — сбор в обучающую выборку *hard negative examples* [31, 32] (фрагментов изображений, внешне похожих на детектируемый объект, но не являющихся детектируемым объектом). Такие объекты нужны, для того чтобы научиться отличать объект от похожих объектов фона. Сложность в том, что в естественных условиях такие объекты встречаются редко, поэтому для эффективного обучения требуются специальные методы майнинга таких объектов. Одним из эффективных приемов майнинга hard negative examples из соревновательной практики *Kaggle* [33] является использование в качестве hard negative examples изображений, вызвавших ложноположительные срабатывания недоученной версии детектора (детектора, полученного после небольшого числа эпох обучения). Данный прием использовался в решении, занявшем первое место в соревновании [34].

Generative Adversarial Networks (GAN). *Generative Adversarial Networks* [35] могут использоваться для генерации изображений [36] либо для стилизации (*Style Transfer* [37]) изображений под новые условия. Современные GAN часто генерируют некорректные изображения, также существуют большие проблемы с их сходимостью, поэтому GAN не всегда подходят для генерации самих распознаваемых объектов, од-

нако их вполне можно использовать для генерации фона [38] либо для адаптации имеющихся изображений к другим условиям (например, для переделывания дневных изображений в ночные [39–41]).

Имитация добавления данных. Широко известный метод регуляризации обучения нейронных сетей *Dropout* [42] можно рассматривать как имитацию добавления данных. С точки зрения i-го слоя нейронной сети нет разницы между тем, изменяются ли входные данные или изменяются значения выходов $i-1$-го слоя нейронной сети. Другой реализацией данного принципа является *Shake-Shake regularization* [43] — метод, некоторое время являвшийся *State-of-the-Art* результатом на *CIFAR-10* [44] — датасете, содержащем малое число изображений (около 10 000), из-за чего имитация добавления данных для этого датасета представляется целесообразной. Развитием идеи Shake-Shake regularization является *Shake-Drop regularization* [45].

Проблема cross-dataset generalization. Алгоритм Cross-Dataset Machine

Одной из проблем, связанных с формированием выборки, является проблема обобщения алгоритма на данные тех же классов из другого датасета — *cross-dataset generalization* [46]. То есть в качестве тестовой выборки используется не выборка из того же датасета, на котором шло обучение, а выборка из другого датасета (допустим, в задаче распознавания машины обучение производилось на ImageNet [22], а валидация — на PASCAL VOC [47]). Cross-dataset обобщающая способность обычно существенно хуже простой обобщающей способности из-за того, что датасеты могут сильно отличаться друг от друга, а в процессе обучения не было обеспечено обобщение на данные, сильно не похожие на обучающие. Cross-dataset тестирование алгоритмов является более надежным в плане определения качества работы алгоритма, так как позволяет исключить влияние на результат тестирования части проблем формирования выборки, таких как ложные внутривыборочные закономерности.

Рассмотрим один из возможных алгоритмов повышения cross-dataset generalization при обучении нейронных сетей (назовем этот алгоритм *Cross-Dataset Machine*). Допустим, есть три датасета (для одной и той же задачи) D_1, D_2, D_3, обучение производится на датасетах D_1 и D_2, тестирование — на датасете D_3. Будем при обучении в четных батчах подавать на вход сети только объекты из датасета D_1, в нечетных — только объекты из датасета D_2. Если для обучения используется n датасетов D_1, D_2, ..., D_n, то разобьем эти датасеты на два непересекающихся множества M_1 и M_2 ($\{D_1...D_m\} \in M_1$, $\{D_{m+1}...D_n\} \in M_2$) и будем при обучении в четных батчах подавать на вход сети только объекты из множества M_1, в нечетных — только объекты из множества M_2. В качестве датасетов могут выступать не датасеты целиком, а отдельные источники данных, из которых был собран датасет (такая схема чаще всего и используется на практике). Идея алгоритма в том, что при обучении сети на батче из датасета D_1 сеть выучивает два типа фич: специфичные для датасета D_1 и не обобщаемые на другие датасеты (назовем такие фичи F_p) и обобщаемые между датасетами (F_g). При подаче на вход сети батча из датасета D_2 необобщаемые фичи F_p будут разрушаться (не будут выживать), а обобщаемые фичи F_g будут выживать и развиваться дальше. Таким образом, после нескольких итераций алгоритма в сети должны остаться и обучиться только хорошо обобщаемые между датасетами фичи.

Проверим алгоритм на задаче соревнования *State Farm Distracted Driver Detection* [48], проводившегося на платформе Kaggle. По изображению водителя изнутри машины надо определить факт отвлечения водителя (курит, ест, разговаривает по телефону). Датасет соревнования составлен из групп изображений, где каждая группа — это изображения одного и того же водителя в одной и той же машине (всего в датасете 26 водителей). Поэтому в качестве датасетов (источников данных) D_1, D_2, ..., D_n будем использовать группы изображений одного и того же водителя (все изображения одного водителя i — это отдельный источник данных D_i). Разделим их на два непересекающихся множества (по 11 водителей в каждом), пять водителей оставим для валидации. В таблице приводятся результаты сравнения работы данного алгоритма и стандартного алгоритма обучения нейросети. Мы видим, что Cross-Dataset Machine показывает ошибку на валидации меньшую, чем у стандартного алгоритма.

Возможной модификацией алгоритма Cross-Dataset Machine является 3-стадийный Cross-Dataset Machine. В этом алгоритме одна итерация обучения состоит из трех стадий: на первой стадии на вход сети подаются обучающие примеры из подмножества датасетов M_1, на второй стадии — из подмножества датасетов M_2, на третьей стадии — обучающие примеры из всей обучающей выборки вперемешку. Полный цикл обучения сети состоит из k таких итераций. Это позволяет сети лучше адаптироваться ко всей генеральной совокупности, в то же время сохраняются преимущества исходного алгоритма, заключающиеся в выживании хорошо обобщаемых между датасетами фич и невыживании плохо обобщаемых фич. Как мы видим из таблицы, данный алгоритм позволяет получить вы-

■ Результаты работы алгоритма Cross-Dataset Machine на задаче State Farm Distracted Driver Detection

■ Cross-Dataset Machine results on State Farm Distracted Driver Detection task

Ошибка	Стандартный алгоритм	Cross-Dataset Machine	3-стадийный Cross-Dataset Machine
Минимальная	1,78	1,70	1,66
В конце обучения	1,85	1,78	1,71
Средняя	2,05	2,04	2,01

игрыш относительно стандартного Cross-Dataset Machine.

Проверим алгоритм на задаче соревнования *HuBMAP — Hacking the Kidney* [49]. В этой задаче требуется отсегментировать клетки функциональных единиц ткани. Всего дано семь больших цельных изображений функциональных единиц ткани, обучение и распознавание происходит по маленьким фрагментам больших изображений (всего 11 473 фрагмента). Поэтому в качестве датасетов (источников данных) D_1, D_2, ..., D_n будем использовать цельные изображения (одно цельное изображение i — это отдельный источник данных D_i). В качестве backbone использовался EfficientNet-B4 [50]. В результате доля корректно отсегментированных пикселей на задаче HuBMAP — Hacking the Kidney следующая:

— стандартный алгоритм — 0,848;

— Cross-Dataset Machine — 0,855.

И в этой задаче Cross-Dataset Machine позволяет получить выигрыш в качестве.

Заключение

В статье сделан обзор возможных проблем формирования обучающей выборки, проведен анализ их влияния на результат обучения, даны рекомендации по их устранению. Рассмотрены наиболее эффективные способы расширения обучающей выборки. Исследована проблема cross-dataset generalization. Предложен алгоритм Cross-Dataset Machine, позволяющий получить выигрыш в cross-dataset generalization.

Литература

1. **Vapnik V. N., Chervonenkis A. Y.** On the uniform convergence of relative frequencies of events to their probabilities. In: *Measures of complexity*. Springer, Cham, 2015. Pp. 11–30.
2. **Gonzalez-Diaz R., Gutiérrez-Naranjo A., Paluzo-Hidalgo E.** Representative datasets: the perceptron case. *arXiv preprint arXiv:1903.08519*, 2019.
3. **Roh Y., Heo G., Whang S. E.** A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2019. https://doi.org/10.1109/TKDE.2019.2946162
4. **Kotsiantis S. B., Zaharakis I., Pintelas P.** Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 2007, vol. 160, no. 1, pp. 3–24.
5. **Krizhevsky A., Sutskever I., Hinton G. E.** ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017, vol. 60, no. 6, pp. 84–90.
6. **Lin T. Y., Goyal P., Girshick R., He K., Dollár P.** Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
7. **Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P.** SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002, vol. 16, pp. 321–357.
8. **He H., Bai Y., Garcia E. A., Li S.** ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.
9. **Longadge R., Dongre S.** Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.
10. **Japkowicz N., Stephen S.** The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 2002, vol. 6, no. 5, pp. 429–449.
11. **Buda M., Maki A., Mazurowski M. A.** A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018, vol. 106, pp. 249–259.
12. **Johnson J. M., Khoshgoftaar T. M.** Survey on deep learning with class imbalance. *Journal of Big Data*, 2019, vol. 6, no. 1, pp. 1–54.
13. **Liu X. Y., Wu J., Zhou Z. H.** Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, vol. 39, no. 2, pp. 539–550.
14. **Enzweiler M., Gavrila D. M.** Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, vol. 31, no. 12, pp. 2179–2195.
15. **Dalal N., Triggs B.** Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893.

16. **Ying X.** An overview of overfitting and its solutions. *Journal of Physics: Conference Series, IOP Publishing*, 2019, vol. 1168, no. 2, pp. 022022.
17. **Browne M. W.** Cross-validation methods. *Journal of Mathematical Psychology*, 2000, vol. 44, no. 1, pp. 108–132.
18. **Lee D. H.** Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on Challenges in Representation Learning*, *ICML*, 2013, vol. 3, no. 2, pp. 896.
19. **Hendrycks D., Mazeika M., Kadavath S., Song D.** Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.
20. **Xie Q.** Self-training with noisy student improves imagenet classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10687–10698.
21. **Pham H., Dai Z., Xie Q., Luong M. T., Le Q. V.** Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020.
22. ImageNet Dataset. http://www.image-net.org/ (дата обращения: 05.01.2021).
23. **Cai Z., Vasconcelos N.** Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. https://doi.org/10.1109/TPAMI.2019.2956516
24. Common Objects in Context (COCO) Dataset. https://cocodataset.org/#home. (дата обращения: 05.01.2021).
25. **DeVries T., Taylor G. W.** Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
26. **Zhong Z., Zheng L., Kang G., Li S., Yang Y.** Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 07, pp. 13001–13008.
27. **Cubuk E. D., Zoph B., Mane D., Vasudevan V., Le Q. V.** Autoaugment: Learning augmentation strategies from data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.
28. **Perez L., Wang J.** The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
29. **Shorten C., Khoshgoftaar T. M.** A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019, vol. 6, no. 1, pp. 1–48.
30. **O'Gara S., McGuinness K.** Comparing data augmentation strategies for deep image classification. *Irish Machine Vision and Image Processing Conference (IMVIP)*, 2019. https://doi.org/10.21427/148B-AR75
31. **Canavet O., Fleuret F.** Efficient sample mining for object detection. *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2014, pp. 48–63.
32. **Jin S. Y., RoyChowdhury A., Jiang H., Singh A., Prasad A., Chakraborty D., Learned-Miller E.** Unsupervised hard example mining from videos for improved object detection. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 307–324.
33. Kaggle: You Machine Learning and Data Science Community. https://www.kaggle.com/ (дата обращения: 05.06.2021).
34. NFL 1st and Future — Impact Detection. https://www.kaggle.com/c/nfl-impact-detection/discussion/209403 (дата обращения: 05.03.2021).
35. **Creswell A., White T., Dumoulin V., Arulkumaran K., Sengupta B., Bharath A.** Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 2018, vol. 35, no. 1, pp. 53–65.
36. **Antoniou A., Storkey A., Edwards H.** Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.
37. **Xu Z., Wilber M., Fang C., Hertzmann A., Jin H.** Learning from multi-domain artistic images for arbitrary style transfer. *arXiv preprint arXiv:1805.09987*, 2018.
38. **Ma Y., Liu K., Guan Z., Xu X., Qian X., Bao H.** Background augmentation generative adversarial networks (BAGANs): Effective data generation based on GAN-augmented 3D synthesizing. *Symmetry*, 2018, vol. 10, no. 12, pp. 734.
39. **Meng Y., Kong D., Zhu Z., Zhao Y.** From night to day: GANs based low quality image enhancement. *Neural Processing Letters*, 2019, vol. 50, no. 1, pp. 799–814.
40. **Ardiyanto I., Soesanti I., Qairawan D. C.** Night-today road scene translation using generative adversarial network with structural similarity loss for night driving safety. In: *Deep Learning and Big Data for Intelligent Transportation: Enabling Technologies and Future Trends*, 2021. Pp. 119–133.
41. **Xie H., Xiao J., Lei J., Xie W., Klette R.** Image Scene Conversion Algorithm Based on Generative Adversarial Networks. In: *Asian Conference on Pattern Recognition*. Springer, Singapore, 2019. Pp. 29–36.
42. **Srivastava N., Hinton G. E., Krizhevsky A., Sutskever I., Salakhutdinov R. R.** Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 2014, vol. 15, no. 1, pp. 1929–1958.
43. **Gastaldi X.** Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.
44. CIFAR-10 Dataset. https://www.cs.toronto.edu/~kriz/cifar.html (дата обращения: 05.02.2021).
45. **Yamada Y., Iwamura M., Akiba T., Kise K.** Shakedrop regularization for deep residual learning. *IEEE Access*, 2019, vol. 7, pp. 186126–186136.
46. **Torralba A., Efros A. A.** Unbiased look at dataset bias. *CVPR*, 2011, pp. 1521–1528.
47. PASCAL VOC Dataset. http://host.robots.ox.ac.uk/pascal/VOC/ (дата обращения: 05.01.2021).
48. State Farm Distracted Driver Detection. https://www.kaggle.com/c/state-farm-distracted-driver-detection (дата обращения: 05.06.2021).

49. HuBMAP — Hacking the Kidney. https://www.kaggle.com/c/hubmap-kidney-segmentation (дата обращения: 05.06.2021).

50. **Tan M., Le Q.** Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning, PMLR*, 2019, pp. 6105–6114.

UDC 004.855.5
doi:10.31799/1684-8853-2021-4-61-70

Training set formation in machine learning tasks. Survey

A. V. Parasich[a], Programmer Engineer, orcid.org/0000-0003-2728-0893, parasichav@yandex.ru
V. A. Parasich[b], PhD, Tech., Associate Professor, orcid.org/0000-0003-3593-2345
I. V. Parasich[b], PhD, Tech., Associate Professor, orcid.org/0000-0003-1965-8737
[a]3DiVi Inc, 64, Lenina Pr., 454080, Chelyabinsk, Russian Federation
[b]South Ural State University, 76, Lenina Pr., 454080, Chelyabinsk, Russian Federation

Introduction: Proper training set formation is the key factor in solving machine learning tasks. At the same time, in real training sets, there are often some problems and errors that have a critical impact on the training result. The training set formation problem arises in all machine learning problems; therefore, knowledge of the possible problems of forming a training set will be useful when solving any machine learning problem. **Purpose:** Make an overview of possible problems in the formation of a training set, in order to facilitate their detection and elimination when working with real training sets. Analyze the impact of these problems on learning. **Results:** The article makes on overview of possible errors in the formation of a training set, such as lack of data, imbalance, false patterns, sampling from a limited set of sources, change in the general population over time, and others. The influence of these errors on the learning result is considered. The influence of the same problems on the formation of a test set and measurement of the quality of the learning algorithm is considered. The pseudo-labeling, data augmentation, hard samples mining are considered as the most effective ways to expand the training set. Practical recommendations for the formation of training and test set are offered. Practical recommendations for the formation of training and test set are offered. Examples from the practice of Kaggle competitions are given. The problem of cross-dataset generalization is considered. An algorithm for solving the problem of cross-dataset generalization in training neural networks, called the Cross-Dataset Machine, is proposed, which is very simple to implement and allows you to get a gain in cross-dataset generalization. **Practical relevance:** The materials of the article can be used as a practical guide in solving machine learning problems.

Keywords — machine learning, training set, Kaggle, deep neural networks, decision trees, ImageNet.

For citation: Parasich A. V., Parasich V. A., Parasich I. V. Training set formation in machine learning tasks. Survey. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2021, no. 4, pp. 61–70 (In Russian). doi:10.31799/1684-8853-2021-4-61-70

References

1. Vapnik V. N., Chervonenkis A. Y. *On the uniform convergence of relative frequencies of events to their probabilities.* In: *Measures of complexity*. Springer, Cham, 2015. Pp. 11–30.
2. Gonzalez-Diaz R., Gutiérrez-Naranjo A., Paluzo-Hidalgo E. Representative datasets: the perceptron case. *arXiv preprint arXiv:1903.08519*, 2019.
3. Roh Y., Heo G., Whang S. E. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2019. https://doi.org/10.1109/TKDE.2019.2946162
4. Kotsiantis S. B., Zaharakis I., Pintelas P. Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 2007, vol. 160, no. 1, pp. 3–24.
5. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017, vol. 60, no. 6, pp. 84–90.
6. Lin T. Y., Goyal P., Girshick R., He K., Dollár P. Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
7. Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002, vol. 16, pp. 321–357.
8. He H., Bai Y., Garcia E. A., Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.
9. Longadge R., Dongre S. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.
10. Japkowicz N., Stephen S. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 2002, vol. 6, no. 5, pp. 429–449.
11. Buda M., Maki A., Mazurowski M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018, vol. 106, pp. 249–259.
12. Johnson J. M., Khoshgoftaar T. M. Survey on deep learning with class imbalance. *Journal of Big Data*, 2019, vol. 6, no. 1, pp. 1–54.
13. Liu X. Y., Wu J., Zhou Z. H. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, vol. 39, no. 2, pp. 539–550.
14. Enzweiler M., Gavrila D. M. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, vol. 31, no. 12, pp. 2179–2195.
15. Dalal N., Triggs B. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893.
16. Ying X. An overview of overfitting and its solutions. *Journal of Physics: Conference Series, IOP Publishing*, 2019, vol. 1168, no. 2, pp. 022022.
17. Browne M. W. Cross-validation methods. *Journal of Mathematical Psychology*, 2000, vol. 44, no. 1, pp. 108–132.
18. Lee D. H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on Challenges in Representation Learning, ICML*, 2013, vol. 3, no. 2, pp. 896.
19. Hendrycks D., Mazeika M., Kadavath S., Song D. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.
20. Xie Q. Self-training with noisy student improves imagenet classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10687–10698.

21. Pham H., Dai Z., Xie Q., Luong M. T., Le Q. V. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020.
22. *ImageNet Dataset*. Avaliable at: http://www.image-net.org/ (accessed 5 January 2021).
23. Cai Z., Vasconcelos N. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. https://doi.org/10.1109/TPAMI.2019.2956516
24. *Common Objects in Context (COCO) Dataset*. Avaliable at: https://cocodataset.org/#home (accessed 5 January 2021).
25. DeVries T., Taylor G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
26. Zhong Z., Zheng L., Kang G., Li S., Yang Y. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 07, pp. 13001–13008.
27. Cubuk E. D., Zoph B., Mane D., Vasudevan V., Le Q. V. Autoaugment: Learning augmentation strategies from data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.
28. Perez L., Wang J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
29. Shorten C., Khoshgoftaar T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019, vol. 6, no. 1, pp. 1–48.
30. O'Gara S., McGuinness K. Comparing data augmentation strategies for deep image classification. *Irish Machine Vision and Image Processing Conference (IMVIP)*, 2019. https://doi.org/10.21427/148B-AR75
31. Canavet O., Fleuret F. Efficient sample mining for object detection. *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2014, pp. 48–63.
32. Jin S. Y., RoyChowdhury A., Jiang H., Singh A., Prasad A., Chakraborty D., Learned-Miller E. Unsupervised hard example mining from videos for improved object detection. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 307–324.
33. *Kaggle: You Machine Learning and Data Science Community*. Avaliable at: https://www.kaggle.com/ (accessed 5 June 2021).
34. *NFL 1st and Future — Impact Detection*. Avaliable at: https://www.kaggle.com/c/nfl-impact-detection/discussion/209403 (accessed 5 March 2021).
35. Creswell A., White T., Dumoulin V., Arulkumaran K., Sengupta B., Bharath A. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 2018, vol. 35, no. 1, pp. 53–65.
36. Antoniou A., Storkey A., Edwards H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.
37. Xu Z., Wilber M., Fang C., Hertzmann A., Jin H. Learning from multi-domain artistic images for arbitrary style transfer. *arXiv preprint arXiv:1805.09987*, 2018.
38. Ma Y., Liu K., Guan Z., Xu X., Qian X., Bao H. Background augmentation generative adversarial networks (BAGANs): Effective data generation based on GAN-augmented 3D synthesizing. *Symmetry*, 2018, vol. 10, no. 12, pp. 734.
39. Meng Y., Kong D., Zhu Z., Zhao Y. From night to day: GANs based low quality image enhancement. *Neural Processing Letters*, 2019, vol. 50, no. 1, pp. 799–814.
40. Ardiyanto I., Soesanti I., Qairawan D. C. *Night-to-day road scene translation using generative adversarial network with structural similarity loss for night driving safety*. In: *Deep Learning and Big Data for Intelligent Transportation: Enabling Technologies and Future Trends*, 2021. Pp. 119–133.
41. Xie H., Xiao J., Lei J., Xie W., Klette R. *Image scene conversion algorithm based on generative adversarial networks*. In: *Asian Conference on Pattern Recognition*. Springer, Singapore, 2019. Pp. 29–36.
42. Srivastava N., Hinton G. E., Krizhevsky A., Sutskever I., Salakhutdinov R. R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 2014, vol. 15, no. 1, pp. 1929–1958.
43. Gastaldi X. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.
44. *CIFAR-10 Dataset*. Avaliable at:https://www.cs.toronto.edu/~kriz/cifar.html (accessed 5 Febriary 2021).
45. Yamada Y., Iwamura M., Akiba T., Kise K. Shakedrop regularization for deep residual learning. *IEEE Access*, 2019, vol. 7, pp. 186126–186136.
46. Torralba A., Efros A. A. Unbiased look at dataset bias. *CVPR*, 2011, pp. 1521–1528.
47. *PASCAL VOC Dataset*. Avaliable at: http://host.robots.ox.ac.uk/pascal/VOC/ (accessed 5 January 2021).
48. *State Farm Distracted Driver Detection*. Avaliable at: https://www.kaggle.com/c/state-farm-distracted-driver-detection (accessed 5 June 2021).
49. *HuBMAP — Hacking the Kidney*. Avaliable at: https://www.kaggle.com/c/hubmap-kidney-segmentation (accessed 5 June 2021).
50. Tan M., Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, *PMLR*, 2019, pp. 6105–6114.