

Тематический выпуск¹

© 2021 г. С.П. АРСЕЕВ (9413serg@gmail.com),
Л.М. МЕСТЕЦКИЙ, д-р техн. наук (mestlm@mail.ru)
(Московский государственный университет им. М.В. Ломоносова)

СКЕЛЕТ СИМВОЛА КАК МОДЕЛЬ СЛЕДА ПЕРА ДЛЯ РАСПОЗНАВАНИЯ ПО ВОССТАНОВЛЕННОЙ ТРАЕКТОРИИ²

Статья посвящена исследованию задачи распознавания рукописного текста и сведению задачи распознавания по изображению текста к задаче распознавания по следу пера при его написании. Предложен метод, основанный на использовании медиального представления, восстанавливающий след пера по изображению рукописного текста. Предложено обоснование метода на основе экспериментального исследования распознавания символов по их изображению, по восстановленному и по истинному следу пера.

Ключевые слова: глубокое обучение, сверточные нейронные сети, рекуррентные нейронные сети, скелет, распознавание символов, распознавание рукописного текста.

DOI: 10.31857/S0005231021110015

1. Введение

Одной из активно исследуемых задач компьютерного зрения является распознавание рукописного текста. Эта задача находит множество практических применений, от автоматической обработки документов до работы с архивами. Методы, используемые для решения этой задачи, можно разделить на два подхода: offline-распознавание и online-распознавание.

Offline-распознавание рукописного текста представляет собой распознавание по изображению. На практике эта задача встречается чаще всего, и способы ее решения достаточно разнообразны: от классических алгоритмов компьютерного зрения с выделением векторов признаков и последующей классификацией до использования искусственных нейронных сетей со сверточной архитектурой, популярность которых начала быстро расти после того, как в 2012 г. сеть AlexNet [1] превзошла конкурирующие методы. В данный момент нейронные сети превосходят альтернативные методы во многих задачах компьютерного зрения. Классические методы распознавания изображений отличаются разнообразием применяемых дескрипторов: здесь могут

¹ Статьи с 3 стр. по 147 стр. являются окончанием тематического выпуска № 10, 2021.

² Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 20-01-00664).

использоваться дескрипторы LBP [2], SIFT [3], SURF [4], HOG [5] и многие другие. Пример подборки классических методов, использующихся для задачи распознавания рукописного текста, можно увидеть в публикации [6], где на примере распознавания индийского письма деванагари исследуются различные классификаторы, оперирующие на гистограмме ориентированных градиентов изображения. Одно из первых применений сверточных нейронных сетей для задачи распознавания символов описывается в [7].

Online-распознавание представляет собой распознавание следа пера в момент написания текста. Так как такой след имеет вид последовательности координат пера, рекуррентные нейронные сети часто применяются в решении этой задачи, пример использования таких сетей для решения задачи распознавания китайских рукописных символов можно увидеть в [8]. Такая задача зачастую проще в решении, чем задача offline-распознавания из-за меньшей размерности данных, но основным ограничением этого подхода является необходимость наличия данных о траектории, что требует использования оборудования для записи данных и существенно ограничивает возможность применять этот подход на практике. Во многих задачах запись следа пера может отсутствовать, тогда для распознавания доступны только изображения текста.

Цель данной статьи заключается в попытке свести задачу распознавания рукописных символов по их изображению (offline-распознавания) к задаче распознавания по траектории пера (online-распознавания) посредством построения по изображению символа возможной траектории пера при его начертании. Такое сведение позволит применять методы online-распознавания ко многим задачам распознавания текста, что позволит пользоваться преимуществами этой группы методов: они не накладывают таких требований на объем обучающей коллекции и могут оказываться более эффективными. Для реконструкции следа пера используется представление символа в виде графовой модели, что позволяет рассматривать также возможность решения задачи с использованием графовых нейронных сетей, которые успешно применяются для детектирования [9] и распознавания различных объектов. Однако первые эксперименты [10] с применением простейших графовых нейронных сетей для распознавания рукописных символов показали отсутствие заметного преимущества таких моделей по сравнению со сверточными нейронными сетями, обрабатывающими изображение.

2. Восстановление следа пера

Основная идея предложенного подхода заключается в том, что для символа строится его модель в формате геометрического графа. Вершины графа близки к положениям пера в определенные моменты при начертании символа, а ребра между вершинами соответствуют отрезкам траектории пера. Полная траектория пера будет представлять собой набор маршрутов в этом графе, где каждый маршрут соответствует куску траектории, написанному без отрыва пера.

2.1. Построение медиального представления

В качестве основы для графа, моделирующего положения пера, используется медиальное представление (скелет) символа на изображении, впервые описанное в [11]. Введем несколько ключевых понятий, также использовавшихся в [10, 12].

Определение 1. Скелетом (медиальным представлением) называется множество центров всех вписанных кругов фигуры [13, раздел 2.4].

В медиальном представлении фигуры сложной формы обычно присутствует множество коротких “шумовых” ветвей, которые могут быть исключены без существенных потерь информации, необходимой для распознавания, а их наличие, наоборот, затрудняет обработку.

Определение 2. Стрижкой скелета или его регуляризацией называется процесс исключения из скелета ветвей, вклад которых в образование формы объекта незначителен [13, 14, глава 10].

В предложенном алгоритме применяется метод стрижки, подробно описанный в [13, раздел 10.3]. Медиальное представление является множеством точек, соединенных отрезками прямых линий и парабол [13, раздел 7.1], но применительно к данной задаче его можно приблизить неориентированным геометрическим графом $S = (V; E)$. Потери при приближении кусков парабол прямолинейными ребрами незначительны из-за малой длины ребер графа. Построение модели производится по алгоритму, подробно описанному в [13]. После бинаризации изображения связные области на его бинарном представлении аппроксимируются многоугольниками [13, главы 5–6], после чего на этих многоугольниках строится морфологическая модель [13, главы 7–10]. Пример выхода алгоритма после регуляризации показан на рис. 1.

Предлагаемый подход основан на предположении, что скелет изображения символа может использоваться в качестве модели следа пера при написании



Рис. 1. Медиальное представление символа (буква “g”).

этого символа. В большинстве случаев вершины скелетного графа расположены в точках, близких к середине штриха, и существенные отклонения от этого возможны только в случае сильного шума на краях штриха, не дающего точно построить модель, или же существенного перекрытия близко расположенных штрихов, вызывающего визуальное слияние нескольких штрихов в один. Поэтому задача восстановления траектории пера (одной из возможных) сводится к построению маршрута (последовательности инцидентных вершин и ребер с возможными повторами) или множества в скелетном графе, в которые в совокупности должны входить все ребра графа. Предложенный алгоритм строит маршрут для каждой связной компоненты графа без разбиения на пересекающиеся штрихи. Таким образом, восстанавливается возможная траектория начертания “без отрыва пера”. В силу архитектуры рекуррентной нейронной сети, используемой в дальнейшем для распознавания, данные о ребрах не подаются на вход этой сети. Поэтому вместо полного маршрута достаточно сохранять только соответствующую ему последовательность вершин, что и будет являться выводом полного алгоритма.

В общей сложности предложенный в статье алгоритм, сводящий задачу offline-расознавания к задаче online-расознавания, состоит из следующих этапов:

1. Построение медиального представления символа в формате геометрического графа;
2. Конструирование вспомогательного графа (далее — метаграфа) по исходному графу;
3. Обход метаграфа;
4. Построение обхода скелетного графа на основе обхода его метаграфа;
5. Генерализация построенного маршрута посредством удаления близко расположенных вершин и ребер.

2.2. Построение вспомогательного графа

Ребра исходного скелетного графа моделируют фрагменты траектории пера, и обход этих фрагментов определяется направлением движения пера на данном участке. Для последовательности смежных вершин (расположенных на штрихе) степени 2 (не развилок и не терминальных) направление обхода определяется только выбором направления движения пера по всей последовательности. Неоднозначность построения маршрута возникает на развилках (т.е. в вершинах степени 3), а отдельным источником неоднозначности является проблема обхода цикла, так как при обходе требуется обойти все ребра цикла, при этом минимизировав количество повторов. Для облегчения задачи такого обхода графа на его основе строится новый вспомогательный граф.

Определение 3. Метаграфом исходного графа называется вспомогательный граф, где каждому циклу из минимального базиса циклов исходного графа ставится в соответствие единственная вершина. Метаграф строится по алгоритму 1.

Определение 4. Вершина V_1 исходного графа соответствует вершине V_2 его метаграфа, если выполняется одно из следующих условий:

- 1. V_1 входит в один цикл исходного графа, которому при построении вершин метаграфа ставится в соответствие V_2 . В этом случае одной вершине исходного графа может соответствовать несколько вершин метаграфа и, наоборот, каждой вершине метаграфа, поставленной в соответствие какому-либо циклу, соответствуют все вершины, входящие в этот цикл;*
- 2. V_1 не входит ни в один цикл исходного графа, а V_2 ставится ей в соответствие при построении вершин метаграфа. В этом случае соответствие имеет взаимно однозначный характер: каждой такой вершине соответствует ровно одна вершина на другом графе.*

Алгоритм 1 (построение метаграфа).

- I. Выделение в исходном графе минимального базиса циклов, т.е. множества циклов, являющегося базисом пространства циклов для графа (любой цикл графа можно представить как комбинацию циклов из этого множества), имеющего при этом минимальную суммарную длину.*
- II. Построение вершин:*
 - 1. Если вершина не входит ни в один из циклов, она переходит в метаграф напрямую, т.е. в метаграф добавляется одна новая вершина, которая ставится в соответствие этой исходной вершине;*
 - 2. Для каждого цикла из базиса в метаграф добавляется ровно одна новая вершина, которая ставится в соответствие этому циклу.*
- III. Построение ребер:*
 - 1. Если обе вершины ребра исходного графа не входят ни в один из циклов, это ребро переходит в метаграф напрямую, т.е. между соответствующими им вершинами проводится ребро той же длины;*
 - 2. Если одна вершина ребра не входит ни в один из циклов, а вторая входит в один или несколько циклов, то в метаграф добавляются ребра той же длины, соединяющие вершину, соответствующую первой вершине, с каждой из вершин, соответствующих этим циклам;*
 - 3. Если обе вершины ребра входят в разные циклы графа (но само ребро не является частью цикла), то в метаграф добавляется ребро той же длины, которое соединяет две вершины метаграфа, соответствующие этим циклам;*
 - 4. Если два цикла исходного графа имеют хотя бы одну общую вершину, то в метаграф добавляется ребро нулевой длины, которое соединяет вершины, соответствующие этим циклам;*
 - 5. Берется минимальное остовное дерево получившегося графа.*

Построение метаграфа по этому алгоритму однозначно, если однозначен выбор минимального базиса циклов исходного графа. Это всегда выполняется для скелета плоской фигуры, где длина ребра определяется как евклидово расстояние между его концами, и вероятность наличия циклов с

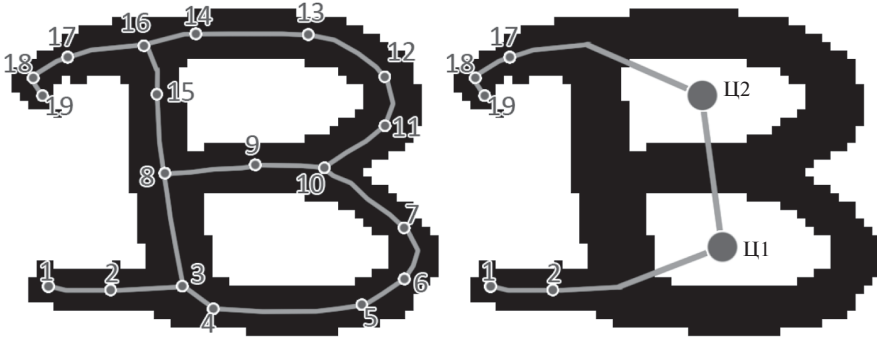


Рис. 2. Пример скелетного графа и его метаграфа.

равной длины мала. Пример построения этого вспомогательного графа показан на рис. 2, где для символа из тестового набора показан его скелетный граф (большинство вершин не показано в целях удобства восприятия) и соответствующий ему метаграф. В этом примере каждый цикл исходного графа превращается в вершину (Ц1 и Ц2) метаграфа, причем вершина Ц1 метаграфа соответствует вершинам 3–10 исходного графа, а вершина Ц2 — вершинам 8–16. Так как циклы могут иметь общие вершины, получающийся на предпоследнем шаге алгоритма 1 граф не обязательно будет являться деревом. При взятии его остовного дерева в качестве метаграфа ни одно ребро, соответствующее ребру исходного графа, не будет потеряно, так как циклы в метаграфе возникают только в результате применения правил III, п. 2 и III, п. 4 из алгоритма построения метаграфа.

2.3. Обход метаграфа

Так как целью восстановления траектории является ее пригодность для распознавания, соответствие реальной траектории не требуется, но желательно. В качестве эвристического критерия для качественной траектории можно предложить минимизацию повторных проходов через ребра графа. Для графа в виде дерева это достигается посредством рекурсивного обхода поддеревьев от наименее глубокого до наиболее глубокого. Этот метод может быть применен для обхода метаграфа, на основе которого уже строится маршрут в исходном графе.

Первый обход метаграфа и его разметка производятся по алгоритму 2.

Алгоритм 2 (обход метаграфа).

I. Выбор начальной вершины:

1. В исходном графе строится список всех вершин степени 1 (конец штриха);
2. Если список пуст (вершин степени 1 в графе нет), то рассматриваются все вершины графа;
3. Для каждой вершины из списка считается ее сумма координат $S = x + y$;

4. Выбирается вершина N_0 из списка с минимальной суммой S ;
5. Если на шаге 2 рассматривались все вершины графа, то в исходный граф добавляется новая вспомогательная вершина N с теми же значениями координат, что и N_0 , соединенная ребром длины 0 с вершиной N_0 . В метаграф добавляется соответствующая ей вершина с теми же координатами, аналогично соединенная с вершиной, соответствующей N_0 . В противном случае $N = N_0$;
6. Вершина N называется стартовой.

II. Обход метаграфа:

1. Метаграф рассматривается как дерево с корнем в вершине N , которая выбирается в качестве текущей вершины;
2. Повторять следующие шаги:
 - Если у текущей вершины есть дочерние вершины, не помеченные как обойденные, спуститься в одну из них, выбранную произвольно (назначить ее текущей вершиной);
 - Иначе, если текущая вершина является терминальной, присвоить ей значение глубины $d = 1$, пометить ее как обойденную и подняться в родительскую вершину;
 - Иначе (если у текущей вершины есть дочерние вершины, но все помечены как обойденные) присвоить ей значение глубины $d = \max(d_{child}) + 1$, где d_{child} – множество меток глубины ее дочерних вершин; пометить ее как обойденную. Если текущая вершина совпадает с вершиной N (обойден весь граф), завершить алгоритм, иначе подняться в родительскую вершину.

Разберем приведенный алгоритм на примере графа с рис. 2. Обход начинается с вершины 17. Метаграф данного графа имеет линейную структуру, что существенно упрощает обход. В качестве стартовой выбирается вершина 19. Алгоритм последовательно спускается в вершину 1 и присваивает ей глубину 1, после чего возвращается, помечая глубину остальных вершин (вершина 2 будет иметь значение 2, вершина Ц1 – значение 3 и т.д.).

2.4. Обход графа

После первого обхода метаграфа производится второй обход, во время которого в соответствии с алгоритмом 3 строится трасса.

Определение 5. Трассой в графе называется последовательность вершин графа, соответствующая последовательности вершин в некотором маршруте, в который входят все ребра и вершины данного графа. Трасса, которая строится данным алгоритмом, соответствует возможной последовательности положений пера при написании символа.

Алгоритм 3 (обход графа).

1. Если текущая вершина не соответствует никакому из циклов исходного графа, то занести ее в трассу и спуститься в еще не обойденную дочернюю вершину с минимальной глубиной.

2. Если текущая вершина соответствует какому-либо циклу исходного графа, то выбрать финальную вершину, где будет производиться выход из цикла. Эта вершина соответствует дочерней вершине с максимальной глубиной. Также выбрать направление обхода цикла. Двигаться по соответствующему циклу исходного графа, заносая в трассу пройденные вершины, от точки входа (из предыдущей вершины) до ближайшей точки выхода, соответствующей еще не обойденной вершине метаграфа и не совпадающей с финальной вершиной выхода. Если не обойдена только финальная вершина, то двигаться по циклу, заносая пройденные вершины в трассу, до этой вершины.

3. Если дочерних вершин нет (вершина является листом дерева) или все они уже обойдены, то занести текущую вершину в трассу и вернуться на уровень вверх. Для вершины, соответствующей циклу исходного графа, это означает движение по циклу до точки изначального входа в цикл (как и ранее, с занесением в трассу всех пройденных вершин).

Ни одно ребро исходного графа не будет потеряно при таком обходе, так как обход покрывает все вершины и ребра метаграфа. Каждому ребру исходного графа, не входящему ни в какой из циклов, соответствует ребро метаграфа (при выделении остовного дерева на последнем этапе построения метаграфа из него исключаются только дубликаты ребер и те ребра, которые не имеют аналогов на исходном графе). Ребра же, которые входят в циклы исходного графа, обходятся при рассмотрении вершины, соответствующей этому циклу в метаграфе. Такой обход позволяет по метаграфу построить трассу в исходном графе.

Помимо минимизации возвратов и повторных проходов, сортировка дочерних вершин на глубине соответствует специфике задачи: так как короткие ответвления будут обходиться в первую очередь, метод применим не только для построения обхода в отдельных символах, но и для более длинных элементов рукописного текста (слитно написанных слов).

На рис. 2 обход исходного графа начинается с вершины 19 и продолжается до прихода в вершину 16, которая соответствует вершине Ц2 метаграфа. При выборе обхода против часовой стрелки алгоритм выбирает точку выхода в вершине 8, после чего последовательно обходит вершины цикла 15, 8, 9, 10, 11 и далее до возврата в вершину 16. Далее производится переход в вершину 8 через вершину 15 и обход цикла Ц1: точка выхода находится в вершине 3, и алгоритм обходит цикл в следующем порядке: 3, 4, ..., 7, 10, 9, 8, 3, после чего обход цикла завершается и обходятся вершины 2 и 1. Итоговая трасса будет иметь вид [19, 18, 17, 16, 15, 8, 9, 10, 11, 12, 13, 14, 16, 15, 8, 3, 4, 5, 6, 7, 10, 9, 8, 3, 2, 1].

2.5. Генерализация трассы

В местах большой кривизны штрихов текста его скелетный граф будет иметь большое количество вершин, расположенных близко одна к другой. В трассе, получающейся при обходе такого скелета, также будет много близко расположенных вершин, идущих подряд. Объединение таких вершин (или исключение из трассы лишних вершин) позволяет существенно уменьшить

объем входных данных без существенных потерь важной информации, описывающей форму символа. Это облегчает обучение классификатора при решении задачи распознавания по траектории. Прореживание трассы производится в соответствии с алгоритмом 4.

Алгоритм 4 (прореживание трассы).

- I. Вершины начинают рассматриваться со второй вершины трассы (так как трасса строится таким образом, что она всегда начинается в терминальной вершине).
- II. Для вершины проверяются следующие критерии:
 1. Вершина имеет степень 2;
 2. Предыдущая или следующая вершина последовательности тоже имеет степень 2;
 3. Расстояние от рассматриваемой вершины до обеих вершин, рассмотренных на предыдущем шаге, не превышает порогового значения.
- III. В случае выполнения всех трех критериев вершина удаляется из последовательности.
- IV. Рассмотрение переходит к следующей вершине.

В результате всех вышеописанных действий из скелетного графа получается трасса, последовательность его вершин. Эта последовательность объявляется реконструированным следом пера (одним из возможных) при написании рассматриваемого рукописного символа или слова.

3. Распознавание символов

Пригодность восстановленной траектории пера для распознавания рукописного символа по нему обосновывается посредством экспериментального исследования работы алгоритма распознавания на этой восстановленной траектории и сравнения с результатами распознавания по реальной траектории. Сведение задачи offline-распознавания к задаче online-распознавания обосновывается также посредством экспериментального исследования и сравнением качества распознавания символов при помощи предложенных алгоритмов и при помощи традиционных методов offline-распознавания.

Для распознавания символов на основе восстановленной траектории применялся алгоритм, описанный в [15] и основанный на архитектуре двунаправленной рекуррентной нейронной сети, которая состояла из 24 управляемых рекуррентных блоков [16]. Рекуррентная нейронная сеть представляет собой вариант искусственной нейронной сети, предназначенной для работы с данными, имеющими формат последовательности. Структурный элемент такой сети, называемый рекуррентным блоком (управляемый рекуррентный блок представляет собой один из типов таких блоков), выводит два значения: вывод рекуррентного блока и скрытое состояние, а на вход принимает также два значения: текущий элемент последовательности и скрытое состояние с обработки предыдущего элемента последовательности. Размерность каждого блока определяет размерность скрытых состояний.

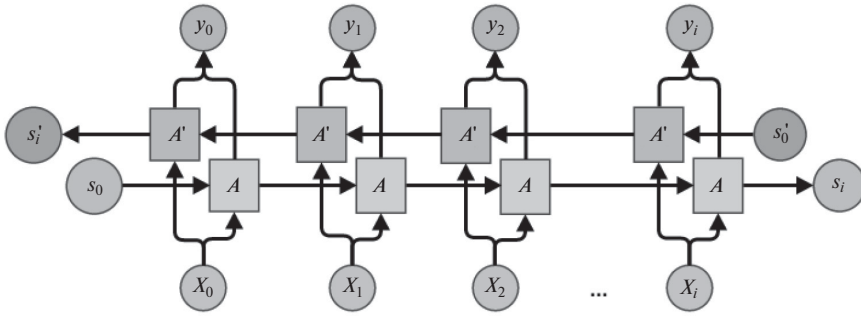


Рис. 3. Двухнаправленная рекуррентная нейронная сеть.

Двухнаправленная рекуррентная нейронная сеть представляет собой две схожие рекуррентные нейронные сети, которые обрабатывают последовательность в двух противоположных направлениях. Выход этих сетей объединяется для получения итогового результата. Схема такой сети показана на рис. 3, на котором A обозначает рекуррентный блок, x – элементы последовательности, S – скрытые состояния, y – вывод для каждого элемента, i – длина последовательности. Итоговый вывод нейронной сети для всей последовательности получается из y_0 и y_i .

Предобучение модели в данном случае не проводилось; обучающая и тестовая выборки имели соотношение размеров 4 к 1. Обучающая выборка также была расширена посредством случайных сдвигов точек исходных последовательностей.

Для сравнения с алгоритмом offline-распознавания использовалась сверточная нейронная сеть VGG16 [17], предобученная на базе ImageNet. Разбиение на обучающую и тестовую выборки проводилось также в соотношении 4 к 1.

4. Экспериментальное исследование

4.1. Малый объем данных

Первый эксперимент проводился на коллекции данных, представленной в [15]. Эта коллекция данных представляет собой траектории пера, записанные с помощью сенсорного контроллера. Набор данных состоит из семи классов и содержит по 100 примеров для каждого класса.

Для получения изображений символов для offline-распознавания эти траектории были растеризованы, дальше предложенный алгоритм работал с этими изображениями без использования данных о реальной траектории пера. Примеры растеризованных изображений показаны на рис. 4.

Результаты экспериментального исследования

Метод	Точность
RNN-True	0,93
RNN-Offline	0,87
VGG	0,60



Рис. 4. Примеры символов, класс “А”.

Результаты экспериментального исследования приведены в таблице. Методам присвоены следующие обозначения:

RNN-True — рекуррентная нейронная сеть, которая была обучена на истинной траектории пера;

RNN-Offline — рекуррентная нейронная сеть, которая была обучена на восстановленной предложенным алгоритмом траектории;

VGG — сверточная сеть VGG16, предобученная на ImageNet и дообученная на изображениях из набора.

Видно, что предложенный метод отстает от online-распознавания по истинным траекториям, что достаточно предсказуемо, но при этом его точность заметно превосходит точность метода offline-распознавания. Сверточная сеть VGG16 оказалась неспособна обучиться на такой выборке, в отличие от рекуррентной модели.

4.2. Большой объем данных

Второй эксперимент проводился на коллекции данных EMNIST [18], представляющем собой преобразованный набор данных NIST Special Database 19 [19]: по сравнению с оригинальным набором данных размер изображений уменьшен, а сами изображения из бинаризованных переведены в оттенки серого. Следует отметить, что предложенный алгоритм, основанный на построении морфологического скелета и восстановлении траектории, устойчив к вышеописанным преобразованиям, так как один из этапов алгоритма включает в себя (повторную) бинаризацию изображения, координаты узлов скелетного графа являются дробными числами, а алгоритм обхода инвариантен к сдвигу и масштабированию координат узлов графа.

Основными трудностями для обучения сверточных нейронных сетей на наборе EMNIST является высокое число классов (10 цифр, 26 прописных букв латинского алфавита и 26 строчных, итого 62 класса), повышенная схожесть отдельных классов между собой (в частности, для некоторых букв строчные и прописные написания практически идентичны друг другу), а также несбалансированность классов в наборе. В связи с этим многие исследователи упрощают постановку задачи, объединяя часть классов между собой. В [20] набор данных был сокращен до 47 классов, итоговый размер обучающей выборки составил 112 800 изображений, а тестовой — 18 800 изображений. На получившемся наборе искусственная сверточная сеть показала точность классификации, равную 81%.

Предложенный алгоритм проверялся на всех классах набора без их объединения и балансирования. Несмотря на простоту предложенной рекуррентной модели, она показала точность классификации, равную 74%, что сопоставимо с точностью сверточной модели, решавшей более простую задачу. Большая часть ошибок алгоритма связана именно с межклассовым сходством, те символы, которые не имеют аналогов со сходным написанием, распознаются с высокой точностью.

5. Заключение

Предложен метод сведения задачи offline-распознавания рукописных символов к задаче online-распознавания, проиллюстрированный посредством использования рекуррентной искусственной нейронной сети для распознавания на двух коллекциях данных. При малом количестве обучающих примеров модель показала способность к обучению на данных, получаемых предложенным алгоритмом, в отличие от сверточной модели, существенно отставшей по точности.

На большом наборе данных алгоритм показал способность к обучению и классификации несмотря на сложность набора данных, в связи с которой большинство работ, использующих этот набор, оперируют упрощенными данными с объединенными классами. С учетом повышенной сложности задачи, показанная алгоритмом точность сопоставима с методами, основанными на сверточных аналогах.

Дальнейшие направления развития этого исследования могут включать в себя эксперименты с совершенствованием способов восстановления траектории пера, а также эксперименты на различных данных, например слитных последовательностях символов.

СПИСОК ЛИТЕРАТУРЫ

1. *Krizhevsky A., Sutskever I., Hinton G.E.* Imagenet Classification with Deep convolutional Neural Networks // *Advances in neural information processing systems*. 2012. P. 1097–1105.
2. *Ojala T., Pietikainen M., Harwood D.* A Comparative Study of Texture Measures with Classification Based on Feature Distributions // *Pattern Recognition*. 1996. V. 29. No. 1. P. 51–59.
3. *Lowe D.G.* Distinctive image features from scale-invariant keypoints // *Inte. J. of Computer Vision*. 2004. V. 16. No. 2. P. 91–110.
4. *Bay H., Ess A., Tuytelaars T., Van Gool L.* SURF: Speeded up Robust Features // *Computer Vision and Image Understanding*. 2008. V. 110. No. 3. P. 346–359.
5. *Dalal N., Triggs B.* Histograms of Oriented Gradients for Human Detection // *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2005)*. 2005. V. 1. P. 886–893.
6. *Pal U., Wakabayashi T., Kimura F.* Comparative Study of Devnagari Handwritten Character Recognition Using Different Feature and Classifiers // *2009 10th Int. Conf. on Document Analysis and Recognition*. 2009. P. 1111–1115.

7. *Ciresan D.C., Meie, U., Gambardella L.M., Schmidhuber J.* Convolutional Neural Network Committees for Handwritten Character Classification // 2011 Int. Conf. on Document Analysis and Recognition. 2011. P. 1115–1139.
8. *Zhang X.Y., Yin F., Zhang Y.M., Liu C.L., Bengio Y.* Drawing and Recognizing Chinese Characters with Recurrent Neural Network // IEEE Trans. Pattern Analysis and Machine Intelligence. 2017. V. 40. No. 4. P. 849–862.
9. *Захаров А.А., Баринов А.Е., Жизняков А.Л., Титов В.С.* Поиск объектов на изображениях с использованием структурного дескриптора на основе графов // Компьютерная оптика. 2018. Т. 42. № 2. С. 283–290.
10. *Ломов Н.А., Арсеев С.П.* Нейронные сети для распознавания формы по медиальному представлению // Тр. Междунар. конф. по компьютерной графике и зрению “Графикон”. ФГУ “Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша РАН”. 2018. № 28. С. 218–221.
11. *Blum H.A.* A Transformation for Extracting New Descriptors of Shape // Proc. Sympos. Models for the Perception of Speech and Visual Form. 1967. V. 4. P. 362–380.
12. *Арсеев С.П., Местецкий Л.М.* Распознавание рукописного текста по восстановленному следу пера с помощью медиального представления // Информационные технологии и нанотехнологии (ИТНТ-2020). 2020. С. 683–689.
13. *Местецкий Л.М.* Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. М.: Физматлит, 2009.
14. *Shaked D., Bruckstein A.M.* Pruning Medial Axes // Computer Vision and Image Understanding. 1998. V. 69. No. 2. P. 156–189.
15. *Fischer T.* “Online Handwritten Character Recognition with capacitive sensors” project // Online: <https://github.com/tobiasfshr/online-handwritten-character-recognition-capacitive-sensors>. 2018.
16. *Chung J., Gulcehre C., Cho KH., Bengio Y.* Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling // arXiv preprint. 2014. arXiv:1412.3555.
17. *Simonyan K., Zisserman A.* Very Deep Convolutional Networks for Large-scale Image Recognition // arXiv preprint. 2014. arXiv:1409.1556.
18. *Cohen G., Afshar S., Tapson J., Van Schaik A.* EMNIST: Extending MNIST to Handwritten Letters // 2017 Int. Joint Conf. on Neural Networks (IJCNN). 2017. P. 2921–2926.
19. *Grother P., Hanaoka K.* NIST Special Database 19 Hhandprinted Forms and Characters 2nd Edition // National Institute of Standards and Technology, Tech. Rep. Online: <http://www.nist.gov/srd/upload/nistSD19.pdf>. 2016.
20. *Neftci E.O., Augustine C., Paul S., Detorakis G.* Event-driven Random Back-propagation: Enabling Neuromorphic Deep Learning Machines // Frontiers in Neuroscience. 2017. V. 11. P. 324.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 20.01.2021

После доработки 15.06.2021

Принята к публикации 30.06.2021