DOI: 10.15514/ISPRAS-2023-35(2)-2

Генерация изображений рукописного текста на русском языке

[1] *А.О. Богатенкова, ORCID: 0000-0001-8679-1568 <nastyboget@ispras.ru>*
[2] *О.В. Беляева, ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>*
[2] *А.И. Перминов, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*

[1] *Московский государственный университет имени М.В. Ломоносова, 119991, Россия, Москва, Ленинские горы, д. 1*
[2] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*

Аннотация. Задача автоматического распознавания рукописного текста является важной составляющей в процессе анализа электронных документов, однако её решение все еще далеко от идеала. Одной из основных причин сложности распознавания рукописного текста на русском языке является недостаточное количество данных, используемых для обучения моделей распознавания. При этом, для русского языка проблема встаёт более остро и усугубляется большим разнообразием сложных почерков. В данной работе исследуется влияние различных методов генерации дополнительных обучающих наборов данных на качество моделей распознавания: метод на основе рукописных шрифтов, метод склейки слов из символов StackMix, метод на основе генеративно-состязательной сети. В рамках данной работы был разработан новый метод создания изображений рукописного текста на русском языке на основе шрифтов. Кроме того, предлагается алгоритм формирования нового кириллического рукописного шрифта на основе имеющихся изображений рукописных символов. Эффективность разработанного метода проверялась с помощью экспериментов, которые проводились на двух общедоступных кириллических наборах данных с помощью двух различных моделей распознавания. Результаты экспериментов показали, что разработанный метод генерации изображений позволил повысить точность распознавания рукописного текста в среднем на 6%, что сравнимо с результатами других, более сложных методов. Исходный код экспериментов, предложенного метода, а также сгенерированные в процессе экспериментов наборы данных выложены в открытый доступ и готовы для скачивания.

Ключевые слова: распознавание рукописного текста; генерация рукописного текста; глубокое обучение; компьютерное зрение

Для цитирования: Богатенкова А.О., Беляева О.В., Перминов А.И. Генерация изображений рукописного текста на русском языке. Труды ИСП РАН, том 35, вып. 2, 2023 г., стр. 19-34. DOI: 10.15514/ISPRAS–2023–35(2)–2

Generation of images with handwritten text in Russian

[1] *A.O. Bogatenkova, ORCID: 0000-0001-8679-1568 <nastyboget@ispras.ru>*
[2] *O.V. Belyaeva, ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>*
[2] *A.I. Perminov, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*

[1] *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia*
[2] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

Abstract. Automatic handwriting recognition is an important component in the process of electronic documents analysis, but its solution is still far from ideal. One of the main reasons for the complexity of Russian handwriting recognition is the insufficient amount of data used to train recognition models. Moreover, for the Russian language the problem is more acute and is exacerbated by a large variety of complex handwriting. This paper explores the impact of various methods of generating additional training datasets on the quality of recognition models: the method based on handwritten fonts, the StackMix method of gluing words from symbols, and the use of a generative adversarial network. A font-based method for creating images of handwritten text in Russian has been developed and described in this work. In addition, an algorithm for the formation of a new Cyrillic handwritten font based on the existing images of handwritten characters is proposed. The effectiveness of the developed method was tested using experiments that were carried out on two publicly available Cyrillic datasets using two different recognition models. The results of the experiments showed that the developed method for generating images made it possible to increase the accuracy of handwriting recognition by an average of 6%, which is comparable to the results of other more complex methods. The source code of the experiments, the proposed method, as well as the datasets generated during the experiments are posted in the public domain and are ready for download.

Keywords: handwritten text recognition; handwritten text generation; deep learning; computer vision

For citation: Bogatenkova A.O., Belyaeva O.V., Perminov A.I. Generation of images with handwritten text in Russian. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023. pp. 19-34 (in Russian). DOI: 10.15514/ISPRAS-2023-35(2)-2

1. Введение

Рукописные записи повсеместно используются в нашей повседневной жизни, как правило, в заметках, списках или других коротких текстах. До изобретения печатного станка в XV веке, рукописи были единственным способом передачи и сохранения информации различного рода. Поэтому огромное количество информации содержится в рукописном виде в исторических документах. Кроме того, рукописный текст систематически используется в других областях, например, в написании конспектов на академических занятиях, на деловых встречах или при заполнении различных бланков и заявлений.

Несмотря на распространенность и удобство использования рукописных записей различного рода, этот способ в настоящее время не является предпочтительным. Основной недостаток рукописей связан с исключительной трудностью их цифровизации с целью более удобного хранения, структуризации и распространения информации. В современном мире большая часть процессов работы с данными автоматизирована, с ними работают компьютеры. Однако компьютер не умеет работать с аналоговыми данными, такими как изображения рукописного текста, эти данные должны быть представлены в понятном для машины виде. В этом контексте способность распознавать и оцифровывать содержимое рукописного текста необходима для извлечения из него необходимой информации.

Автоматическое распознавание рукописного текста (handwritten text recognition, HTR) – задача, которая решается в течение уже довольно продолжительного времени. Она состоит в автоматическом переводе изображений, содержащих рукописный текст, в символьное представление. Эту задачу можно сформулировать следующим образом: пусть $x^{m \times n \times c}$ –

входное изображение шириной m, высотой n и числом каналов c; $y^t = (y_1 \dots y_t), y_i \in A, i = 1, \dots, t$ – выходная последовательность символов из алфавита A; $X = \{x^{m \times n \times c}, m, n > 0, c \in \{1,3\}\}, Y = \{y^t, 0 < t \leq T\}$ – множества входных изображений и выходных последовательностей соответственно. Задача распознавания рукописного текста состоит в определении отображения: $X \to Y$, задающего для каждого изображения рукописного текста его цифровое представление в виде последовательности символов. В настоящее время задача в общем виде ещё не решена и активно исследуется.

Отдельно следует сказать о распознавании рукописного текста на русском языке. Большинство исследований в области распознавания ведется для текста на английском языке, либо на языке, основой которого являются латинские символы. Вследствие этого намного проще найти и наборы данных, и методы решения задачи для текста на таких языках.

При этом существует лишь несколько работ [1, 2], посвящённых распознаванию текста на кириллице, равно как и небольшое количество эталонных наборов данных, используемых для сравнения результатов с другими методами. Всё это добавляет дополнительные сложности на пути решения задачи к существующим многочисленным проблемам.

Как правило, для решения задачи распознавания рукописного текста применяются методы машинного и глубокого обучения, требующие большого объема разнообразных обучающих данных. Так, одной из основных причин отсутствия хорошей универсальной модели распознавания рукописного текста является то, что размер обучающих данных недостаточно велик. В этом случае к существующим данным применяют методы аугментации данных, в рамках рассматриваемой задачи мы остановимся на рассмотрении подвида аугментации – генерации дополнительных синтетических данных.

Как правило, расширение обучающего набора данных осуществляется путем генерации изображений рукописного текста с помощью рукописных шрифтов. Эта методика характерна для англоязычных текстов, для которых создано большое количество разнообразных шрифтов. Несмотря на относительную простоту реализации этого метода, в литературе не упоминается его использование в контексте русского языка. Это может быть связано с меньшим разнообразием доступных шрифтов, а также слабой изученностью данной темы в принципе. Помимо метода генерации данных с помощью шрифтов, применяют также и другие техники, зачастую требующие обучения специализированных моделей машинного обучения [2, 3]. Изучение эффективности данных методов может позволить улучшить качество обучаемых моделей распознавания рукописного текста, в частности, на русском языке.

Статья организована следующим образом: разд. 2 содержит информацию об общеизвестных эталонных наборах данных, используемых в рамках задачи распознавания рукописного текста на русском языке, а также описание стандартных метрик оценки качества распознавания и существующие методы генерации дополнительных наборов данных для обучения моделей. В разд. 3 описывается предлагаемый метод генерации синтетических изображений рукописного текста на основе шрифтов, а также полуавтоматический метод создания нового шрифта. Разд. 4 содержит детали экспериментальной проверки предложенного метода и его сравнения с существующими, а разд. 5 – описание полученных результатов. Наконец, в разд. 6 представлены краткие выводы о проделанной работе и предлагаются возможные варианты дальнейших исследований.

2. Обзор существующих методов

В этом разделе описываются существующие открытые наборы данных, используемые в рамках решения задачи распознавания рукописного текста на русском языке. Кроме того, приводятся общепринятые метрики оценки качества моделей распознавания, а также методы, применяемые для расширения обучающих наборов данных.

2.1 Наборы данных на русском языке

Существует несколько общедоступных наборов данных на русском языке для обучения моделей и сравнения результатов. В настоящее время известны два набора данных со словами и предложениями на кириллице:

- Cyrillic Handwriting Dataset [4];
- HKR [5].

Основная информация о наборах данных представлена в табл. 1.

Табл. 1. Описание наборов данных с кириллицей
Table. 1. Cyrillic datasets description

	Cyrillic Handwriting Dataset	**HKR**
Описание	Набор русских текстов длиной не больше 40 символов, собранный из различных Интернет-ресурсов	Набор из русских (95%) и казахских (5%) слов и предложений: ключевые слова, поэмы и алфавит
Размер	train=72286, test=1544	train=45470, val=9359, test1=5057, test2=5057
Уникальные слова	37519	2808
Уникальные почерки	880	–
Фон	Разнообразный фон, встречаются пятна, линии, соседний текст и т.д.	Однообразный светлый фон

Набор данных **Cyrillic Handwriting Dataset** был опубликован в 2022 году, поэтому еще нет работ, содержащих результаты его обработки. Этот набор очень интересен с точки зрения разнообразия данных: в него входят студенческие конспекты, заполненные формы, электронные рукописные документы. Изображения содержат различного рода шумы и неоднородный фон, в некоторых случаях на изображение одного слова попадают части другого или разлиновка листа. Согласно табл. 1, набор данных состоит из более 70 тысяч обучающих примеров – изображений слов и предложений, что также является его существенным достоинством. Кроме того, он является открытым как для научных исследований, так и для коммерческого использования. Примеры изображений, встречающиеся в наборе данных, представлены на рис. 1.

Рис. 1. Примеры изображений набора Cyrillic Handwriting Dataset
Fig. 1. Examples of the Cyrillic Handwriting Dataset dataset images

Наиболее популярным набором данных на русском языке, упоминаемым в научной литературе, является набор казахских и русских слов и предложений **HKR**. Всего в нем содержится более 60 тысяч изображений слов и предложений, написанных примерно 200 различными почерками. Он создавался путем заполнения однотипных форм, поэтому имеет одну особенность – в нем содержится большое количество копий одного и того же текста,

написанного разными почерками. Отдельно авторами [5] дается разбиение набора данных на тренировочный, валидационный и два тестовых набора. Первый тестовый набор содержит слова, которых нет в тренировочном наборе, но написанные почерками, присутствующими в тренировочном наборе. Напротив, второй тестовый набор содержит слова, которые есть в тренировочном наборе, но написанные “новыми” почерками. Эта особенность позволяет провести анализ того, на что в большей степени обращает внимание обучаемая модель: новые способы написания символов или новые сочетания.

Несмотря на значительный объем набора, изображения в нем достаточно хорошего качества и относительно однообразны. Таким образом, произвольное изображение рукописного текста из “реального мира” совершенно не похоже на то, что содержится в описанном наборе данных. В добавление к этому, использование данного набора ограничивается научными исследованиями, для его использования в коммерческих целях необходимо обратиться к его авторам. Примеры изображений, встречающиеся в наборе данных, представлены на рис. 2.

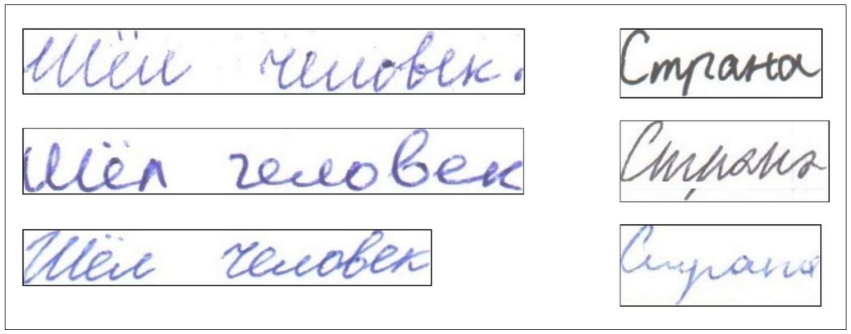

Рис. 2. Примеры изображений набора HKR
Fig. 2. Examples of the HKR dataset images

Таким образом, суммарный размер вышеописанных наборов не превышает 140 тысяч изображений рукописных слов и предложений, а уникальных текстов – не более 40 тысяч, что сигнализирует о недостатке обучающих данных, которые необходимы в большем количестве в силу сложности задачи. Особое внимание следует уделить разнообразию текстов, которое можно увеличить с помощью методов генерации синтетических изображений рукописного текста.

2.2 Метрики оценки качества

Двумя основными метриками, обычно используемыми для оценки моделей распознавания рукописного текста на уровне слов и строк, являются **частота ошибок символов** (Character Error Rate, CER) и **частота ошибок слов** (Word Error Rate, WER).

CER измеряет расстояние Левенштейна [6] между предсказанной и реальной последовательностью символов слова. Расстояние Левенштейна, также иногда называемое расстоянием редактирования, представляет собой метрику для измерения разницы между двумя последовательностями символов. Неформально, расстояние Левенштейна между двумя словами (предсказание модели и реальное слово) – это минимальное количество вставок, удалений или замен, необходимых для преобразования предсказания в правильное слово, делённое на длину правильного слова, как показано в формуле (1):

$$CER(prediction, real) = \frac{substitutions + insertions + deletions}{len(real)}. \quad (1)$$

Частота ошибок в словах (WER) определяется аналогично CER путем вычисления минимального количества вставок, замен и удалений слов, необходимых для перехода от текстовой строки, предсказанной моделью, к реальной текстовой строке.

В некоторых работах наравне с частотой ошибок символов и частотой ошибок слов используется **точность** (accuracy). Данная метрика используется для любых текстовых строк, содержащих как слова, так и предложения.

Точность описывается формулой (2), в которой под равенством подразумевается полное совпадение двух строк:

$$accuracy(prediction, real) = \frac{\sum_{i=1}^{N} pred_i = real_i}{N}. \quad (2)$$

Данная метрика позволяет оценить качество модели более грубо, так как ошибка в одном символе сильно понижает результирующее значение.

2.3 Методы генерации обучающих наборов данных

Как отмечалось в разд. 1, генерация дополнительного набора данных для обучения моделей распознавания является одной из подзадач, которые возникают в рамках задачи распознавания рукописного текста. Метод генерации новых слов и стилей написания в рукописных текстах вряд ли позволит получить по-настоящему реалистичные и совершенно новые изображения, однако он может дать существенный прирост в качестве результатов обучаемых моделей.

Один из интересных способов генерации новых слов и предложений в стиле уже имеющихся слов в наборе данных, предложили авторы статьи [2], которые также предложили метод аугментации путём рисования пятен с помощью кривых Безье. Свой метод они назвали StackMix и состоит он в разбиении слов на символы или подслова, а затем составлении из этих кусочков новых слов. Пример работы данного метода представлен на рис. 3. Такой метод не решает проблему соединения символов и создания новых стилей, зато он может помочь в создании рукописных слов, которых нет в обучающем наборе данных. Тем не менее, метод Stackmix позволил повысить точность распознавания предлагаемой авторами модели с 71% до 80% на наборе HKR.

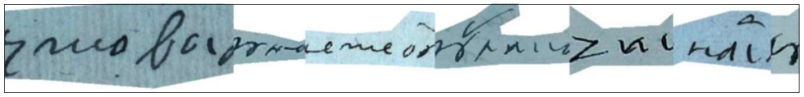

Рис. 3. Пример работы алгоритма Stackmix
Fig. 3. The Stackmix algorithm work example

Следующий метод [7] состоит в генерации изображений рукописного текста с использованием типографских шрифтов. Авторы отобрали 90 тысяч уникальных английских слов и сгенерировали 90 миллионов изображений слов с помощью 750 рукописных шрифтов. Подобный набор слов можно использовать для предобучения моделей распознавания рукописного текста, однако в силу его однообразности потребуется дообучение на реальных данных, чтобы предотвратить переобучение сети. Пример генерации изображения рукописного текста с помощью шрифтов приведен на рис. 4.

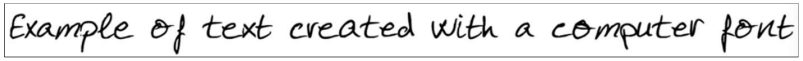

Рис. 4. Пример рукописного текста, полученного с помощью шрифта
Fig. 4. An example of the handwritten text made using a font

В настоящее время активно исследуется целый класс методов генерации рукописного текста, основанных на использовании генеративно-состязательных сетей (GAN) [8]. Такие модели в своей основе содержат две нейронных сети: одна из них генерирует примеры (генеративная модель), а другая пытается отличить подлинные образцы от сгенерированных первой сетью

(дискриминативная модель). Наиболее свежими примерами генеративно-состязательных сетей в области генерации изображений рукописного текста являются ScrabbleGAN [3], GANwriting [9] и TextStyleBrush [10]. Пример результата работы ScrabbleGAN приведен на рис. 5.

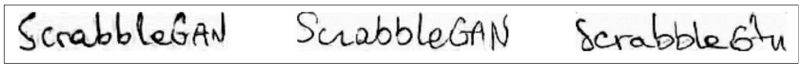

Рис. 5. Пример рукописного текста, полученного с помощью ScrabbleGAN
Fig. 5. An example of the handwritten text made using ScrabbleGAN

Опишем более подробно перечисленные архитектуры генеративно-состязательных сетей для генерации изображений рукописного текста.

- ScrabbleGAN [3] – сеть, состоящая из генератора, дискриминатора и распознавателя символов (OCR). В данной архитектуре дискриминатор влияет на качество изображения, а распознаватель на читаемость текста на изображении. Для обучения сеть использует картинки с текстом и сам текст, стили написания текста меняются с помощью вектора шума, на который домножается входной вектор закодированного текста.
- GANwriting [9] обладает более сложной архитектурой, в которой присутствуют генераторы текста и стиля, соответственно используются дискриминатор, классификатор стиля и распознаватель текста. Вместо одной картинки с текстом, на вход подается группа картинок с текстом, которая формирует конкретный стиль. Поэтому сеть позволяет сгенерировать текст в разных стилях, которые можно смешивать друг с другом.
- TextStyleBrush [10] – наиболее свежая модель, состоящая из 7 нейронных сетей и принимающая на вход картинку со стилем и текст. Здесь также есть генераторы текста и стиля, дискриминатор, классификатор стиля и OCR-модель. Дополняется этот набор сетью, преобразующей выход генератора стиля в набор векторов, а также генератором стилизованного текста, который использует и объединяет результаты работы предыдущих генераторов. Данная модель очень сложна по своей структуре и способна генерировать изображения только тех стилей, на которых была обучена.

Описанные нейросетевые модели генерации обучались и тестировались для английского языка. Соответственно, помимо недостатка, связанного с искусственностью получаемых изображений, имеется и другой – трудоемкость задачи переобучения сети для другого языка. Более того, далеко не все авторы делятся исходным кодом описанных ими моделей.

Таким образом, в научной литературе рассматривается ряд методов, позволяющих увеличить размер обучающего набора данных путём генерации синтетических изображений рукописного текста. При этом лишь один из них [2] был изучен в контексте русского языка, а наиболее простой в реализации и ресурсоёмкости метод на основе шрифтов, насколько нам известно, не применялся с русскими текстами. Поэтому актуально более детальное исследование методов генерации данных применительно к задаче распознавания рукописного текста на русском языке.

3. Описание метода

Согласно разд. 2, метод генерации изображений рукописного текста на основе шрифтов может быть полезен при обучении моделей распознавания, однако он не был использован для русского языка по разным причинам: предположительно, из-за слабой изученности темы, недостатка общедоступных рукописных шрифтов и однообразия получаемых на выходе данных.

Генерация изображений текста с помощью рукописных шрифтов – наиболее простой способ создания дополнительного обучающего набора данных. Этот метод не требует обучения дополнительных моделей, не зависит от какого-либо фиксированного набора данных, прост в реализации и позволяет генерировать данные с большой скоростью. Несмотря на

многочисленные достоинства, генерация текста с помощью шрифтов обладает рядом недостатков. Типографские шрифты, даже рукописные, не обладают достаточным разнообразием и выглядят искусственно не только для нейронной сети, но и для человека.

Как правило, в шрифтах содержится ограниченное количество доступных для написания символов, что влечет за собой зависимость метода от того языка, для которого генерируются дополнительные данные.

Ещё одним существенным недостатком является сложность добавления нового стиля написания, т. е. непосредственно шрифта, так как, как правило, для этого требуется много времени и умение работать со специализированным программным обеспечением. Этот недостаток сильно заметен по отношению к кириллическим шрифтам – существует не так много открытых рукописных шрифтов, содержащих русские символы.

Поэтому имеет смысл каким-то образом автоматизировать процесс создания нового шрифта.

В этом может помочь приложение *calligraphr* [11], позволяющее из изображений символов получить шрифт в формате ttf. Изображения рукописных символов можно взять из базы рукописных символов [12], в которой для каждого рукописного кириллического символа существует более 1500 вариантов написания.

Приложение *calligraphr* требует заполнения шаблона с заранее выбранным набором символов. Шаблон представляет собой изображение, содержащее таблицу, в которой в определённых ячейках должны помещаться изображения конкретных символов.

Примеры пустого и заполненного шаблона представлены на рис. 6.

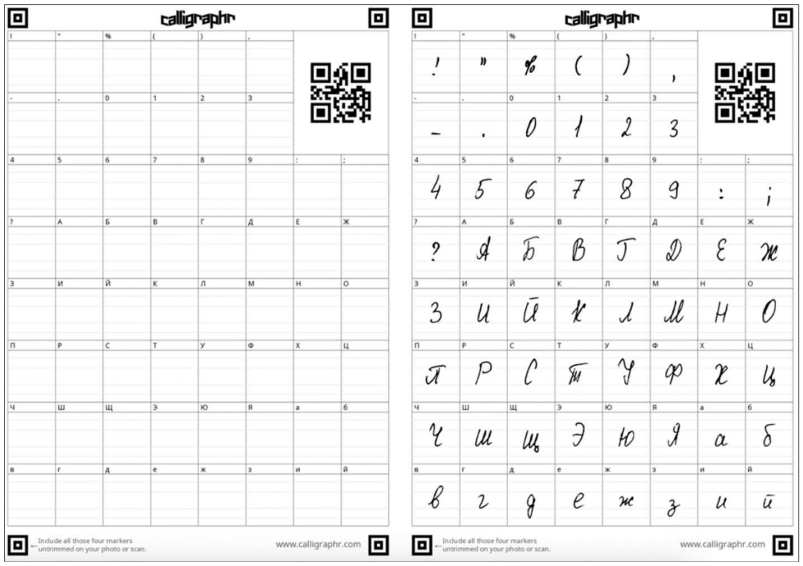

Рис. 6. Примеры пустого и заполненного шаблонов для приложения calligraphr
Fig. 6. Examples of the empty and filled templates for the calligraphr application

Заполненный шаблон далее анализируется приложением, которое извлекает глифы – векторные изображения символов. У полученных глифов можно вручную поменять расположение и размер, добавляя таким образом реалистичность к результату. Затем из глифов формируется шрифт необходимого формата. Таким образом, создание шрифта можно определенным образом автоматизировать.

Для создания нового кириллического шрифта предлагается следующая последовательность действий:

1) заполнение шаблона для *calligraphr* изображениями кириллических символов;
2) ручная коррекция полученных с помощью *calligraphr* глифов при необходимости;
3) итоговая сборка шрифта.

Последние два пункта алгоритма выполняются непосредственно с помощью приложения. Первый пункт требует анализа границ шаблона для поиска координат вставки изображений, а также нахождения базовой линии символа для того, чтобы вставить его в соответствии с ней. Границы можно найти вручную один раз и зафиксировать для конкретного шаблона. Нахождение базовой линии символа можно с некоторыми оговорками реализовать аналогично нахождению базовой линии слова, для этого существует метод на основе устойчивой регрессии, описанный в работе [13]. В силу того, что для символов базовую линию находить сложнее, требуется дополнительная коррекция на втором шаге алгоритма. Пример одного из шрифтов, полученных с помощью описанного выше алгоритма, представлен на рис. 7.

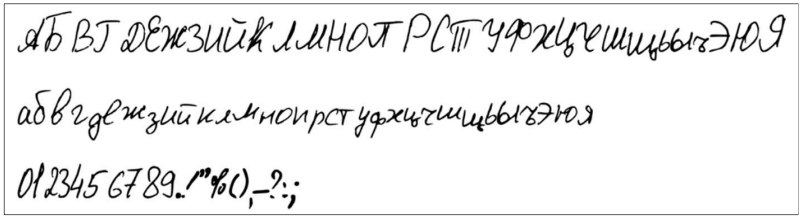

Рис. 7. Пример шрифта, сгенерированного с помощью приложения calligraphr
Fig. 7. An example of the font made using the calligraphr application

Помимо создания новых шрифтов, можно использовать немногочисленные доступные кириллические рукописные шрифты. Среди общедоступных кириллических шрифтов можно выделить следующие: Abram, Anselmo, Benvolio, Capuletty, Gogol, Lorenco, Pushkin, Voronov и другие, доступные на сайтах https://localfonts.eu/freefonts/handwritten-cyrillic-free-fonts, https://www.rufonts.ru/fonts/rukopisnyj, https://allbestfonts.com/category/russian-handwritten, https://fontesk.com/font/handwritten/?tag=cyrillic, https://www.abstractfonts.com/language/9, https://myskotom.ru/shriftoteka. При наличии нескольких десятков шрифтов можно реализовать генератор рукописного текста с помощью многочисленных библиотек отрисовки текста на изображении (например, библиотека Pillow для языка программирования Python), и далее создавать достаточно разнообразные синтетические изображения рукописного текста. Более того, к имеющимся возможностям можно добавить некоторую рандомизацию, напоминающую аугментацию изображений. При создании изображений можно использовать следующие аугментации:

- изменение размера шрифта, сжатие изображения;
- шумы различных видов (Гауссовский, ISO, мультипликативный);
- размытие различных видов (движения, медианное);
- морфологические операции (эрозия, дилатация) для изменения толщины символов;
- изменение наклона шрифта согласно алгоритму из работы [sueiras2021continuous];
- искажение перспективы изображения;
- небольшие обрезка и повороты изображения;
- имитация рукописных зачеркиваний, описанная в работе [shonenkov2021stackmix];
- вставка обрезанных символов по краям изображения (имитация соседних строк);
- добавление случайных (светлых, темных) пятен на изображение;
- изменение яркости, контрастности, насыщенности изображения.

Таким образом, разработан метод генерации кириллических шрифтов, а также изображений рукописного текста на основе шрифтов. Примеры результатов работы генератора рукописного текста представлены на рис. 8.

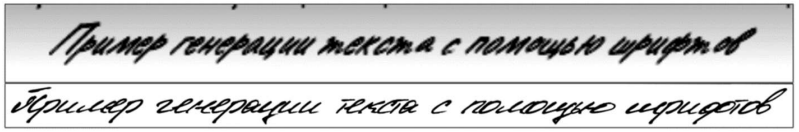

Рис. 8. Примеры результатов работы генератора рукописного текста
Fig. 8. Examples of the handwriting generator's results

4. Описание экспериментов

Для проведения экспериментов зафиксируем модели распознавания рукописного текста, используемые для тестирования методов генерации дополнительных обучающих наборов.

- AttentionHTR [14] – модель архитектуры seq2seq [15], состоящая из сверточного модуля извлечения признаков ResNet [16], рекуррентного модуля разметки последовательности BiLSTM [17] в качестве энкодера, и модуля внимания [18] в качестве декодера. В работе [14] модель была обучена для текстов на английском языке и показала одни из лучших результатов на наборе данных IAM [19].
- Модель с архитектурой трансформер [20] – изначально обучалась на данных английского и русского языка, состоит из свёрточного модуля извлечения признаков ConvNext [21], трансформер-энкодера и двух декодеров: CTC [22] и трансформер [23]. Авторы модели не публиковали подробных результатов на конкретных наборах данных, однако архитектура модели интересна в силу того, что трансформеры являются многообещающим способом решения задачи.

Так как для вышеперечисленных моделей были определены специальные методы стандартизации размера изображений, а также параметры обучения, необходимые для получения наилучшего результата, было решено работать с ними без изменений. При обучении использовались следующие гиперпараметры:

- для AttentionHTR: размер входного изображения 32×100; размер выходного слоя ResNet 512; размер скрытого слоя BiLSTM 256; количество эпох 50; оптимизатор AdaDelta со скоростью обучения $\nu = 1$ и константой затухания $\rho = 0.95$; кросс-энтропия в качестве лосс-функции;
- для трансформера: размер входного изображения 96×384; размер выходного слоя ConvNext 768; количество эпох 100; оптимизатор MADGRAD со скоростью обучения $1 \cdot 10^{-6}$, настраивающейся с помощью CosineAnnealingLR; сумма кросс-энтропии и CTC-лосса в качестве лосс-функции.

Выбранные модели обучались в течение указанного количества эпох, однако обучение могло остановиться ранее, если в течение 10 эпох не уменьшается значение лосс-функции или метрики CER на валидационном наборе данных. Кроме того, процесс обучения организован так, что на каждой итерации берется одинаковое количество данных из каждого набора для предотвращения несбалансированности обучающих данных.

Изображения, используемые для обучения, подавались на вход моделям без проведения предобработки, так как она может потенциально привести к ухудшению их качества. Также была зафиксирована аугментация данных (искажение существующих изображений):

- адаптивное выравнивание гистограммы с ограниченным контрастом (CLAHE);
- небольшие повороты изображения;
- удаление небольших регионов изображения (Cutout);

- искажение сетки (Grid Distortion);
- размытие изображения;
- сжатие JPEG.

Также зафиксируем методы генерации синтетических данных для их сравнения между собой.

- С помощью рукописных шрифтов (разд. 3). В результате поисков готовых шрифтов и создания новых с помощью базы символов было получено 65 различных рукописных шрифтов. Данные, дополняющие наборы HKR и Cyrillic, создавались похожим образом с отличием в количестве преобразований, применяемых к отрисованным изображениям. Для набора, дополняющего HKR, не использовались преобразования, связанные с добавлением пятен на фон, так как изображения набора отличаются белым фоном без шумов. Для набора, дополняющего Cyrillic Handwriting Dataset, использовались всевозможные преобразования, описанные в главе 3, в силу большого стилевого разнообразия изображений, входящих в набор.
- С помощью метода Stackmix [2]. Для использования метода на каждом из наборов HKR и Cyrillic была обучена нейронная сеть, предложенная авторами алгоритма, с рекомендуемыми ими параметрами обучения в течение 100 эпох.
- С помощью генеративно-состязательной сети ScrabbleGAN [3]. На каждом наборе была обучена сеть ScrabbleGAN с параметрами, рекомендуемыми авторами исходного кода, в течение 100 эпох. После этого обученным моделям был переданы корпусы текстовых строк, для которых были сгенерированы синтетические данные, расширяющие наборы HKR и Cyrillic Handwriting Dataset.

Каждый из данных методов применялся к фиксированному набору из 300,000 слов и наборов слов, взятых из случайных статей Википедии [24]. Выбранные тексты статей были очищены от символов, не встречающихся в имеющихся наборах данных, а также были оставлены только уникальные элементы. Для наборов данных HKR и Cyrillic Handwriting Dataset были выбраны различные корпусы текстов в силу того, что наборы символов в них сильно варьировались.

Таким образом, выбраны два кириллических набора данных, три метода расширения этих наборов, а также две модели распознавания рукописного текста. Данные модели были обучены на каждом из наборов без добавления каких-либо данных, а также на наборах, дополненных тремя перечисленными выше способами.

5. Результаты

В табл. 2, 3, 4 и 5 представлены результаты экспериментов, описанных в главе 4. Табл. 2 и 3 содержат результаты обучения модели AttentionHTR, а табл. 4 и 5 – модели трансформер. Табл. 3 и 5 отражают более детальные результаты для набора данных HKR, в котором присутствуют два тестовых набора, описанных в разд. 2.

Табл. 2. Результаты обучения нейронной сети AttentionHTR
Table. 2. The AttentionHTR results

	HKR all			Cyrillic Handwriting Dataset		
	ACC	**CER**	**WER**	**ACC**	**CER**	**WER**
Без генерации	63.67	22.55	37.64	37.95	18.53	60.63
Шрифты	64.69	13.55	31.78	49.09	13.12	49.23
StackMix	**68.96**	**8.45**	**25.04**	**49.9**	**12.44**	**46.03**
GAN	64.69	12.09	30.28	37.69	16.68	59.1

Табл. 3. Результаты обучения нейронной сети AttentionHTR на наборе HKR
Table. 3. The AttentionHTR results on the HKR dataset

	HKR test1			HKR test2		
	ACC	**CER**	**WER**	**ACC**	**CER**	**WER**
Без генерации	39.12	42.1	67.41	**87.86**	3.31	**8.33**
Шрифты	42.32	24.08	55.23	86.73	**3.2**	8.69
StackMix	**53.08**	**13.64**	**40.13**	84.61	3.34	10.2
GAN	43.11	21.08	51.65	85.96	3.24	9.24

Рассмотрим результаты обучения модели AttentionHTR. Исходя из значений метрик, показанных в табл. 2 и 3, можно сделать вывод, что лучшие результаты по части повышения качества обучения модели AttentionHTR показывает алгоритм StackMix, позволяя повысить точность распознавания (accuracy) на 5% и 12% для наборов HKR и Cyrillic Handwriting Dataset соответственно. При этом, если рассматривать набор Cyrillic Handwriting Dataset, то можно заметить, что результаты обучения на наборе, дополненном изображениями, сгенерированными с помощью шрифтов, не сильно хуже результатов для Stackmix – здесь также наблюдается прирост точности распознавания на 12%.

В дополнение к этому, для набора HKR результаты расширения данных с помощью генеративно-состязательной сети ScrabbleGAN практически аналогичны результатам расширения данных с помощью шрифтов, и способ на основе шрифтов явно выигрывает у ScrabbleGAN на наборе Cyrillic Handwriting Dataset. Следует заметить, что использование набора данных, созданного с помощью генеративно-состязательной сети, очень слабо повлияло на улучшение качества распознавания для обоих рассматриваемых наборов данных.

Табл. 4. Результаты обучения нейронной сети трансформер
Table. 4. The transformer's results

	HKR all			Cyrillic Handwriting Dataset		
	ACC	**CER**	**WER**	**ACC**	**CER**	**WER**
Без генерации	66.68	10.56	29.08	48.57	11.71	44.87
Шрифты	73.94	6.29	21.55	52.59	9.58	42.69
StackMix	**76.12**	**4.91**	**18.32**	**59.00**	**8.35**	**37.27**
GAN	72.18	7.49	23.51	49.09	12.26	47.75

Табл. 5. Результаты обучения нейронной сети трансформер на наборе HKR
Table. 5. The transformer's results on the HKR dataset

	HKR test1			HKR test2		
	ACC	**CER**	**WER**	**ACC**	**CER**	**WER**
Без генерации	45.08	19.24	53.33	87.96	2.02	**5.22**
Шрифты	56.18	10.85	37.85	**91.43**	**1.8**	5.5
StackMix	**61.47**	**7.87**	**30.82**	90.56	2.01	6.02
GAN	53.00	13.05	41.68	91.07	2.03	5.62

Рассмотрим результаты обучения модели с архитектурой трансформер. Результаты, показанные в табл. 4 и 5, содержат более высокие показатели, в некотором роде схожие с результатами для AttentionHTR. Здесь также стабильно лидирует алгоритм Stackmix, с помощью которого оказалось возможным поднять точность распознавания (accuracy) примерно на 10%. Кроме того, данные таблиц позволяют сделать вывод о том, что метод генерации синтетических изображений с помощью шрифтов занимает второе место, позволяя поднять точность распознавания на 4–7%. Хуже всех с задачей справился алгоритм

генерации изображений с помощью генеративно-состязательной сети; для набора данных Cyrillic Handwriting Dataset практически не наблюдается каких-либо улучшений в качестве распознавания модели.

В процессе анализа таблиц с результатами обучения моделей (табл. 3 и 5) всплывает еще одна особенность, связанная с набором данных HKR. Как говорилось ранее, тестовый набор HKR делится на две части, обладающие определенными особенностями, которые оказывают сильное влияние на результаты, представленные в таблицах. Тестовая часть test1 содержит новые слова и почерки, на которых модель обучалась, а часть test2 содержит новые почерки, но старые слова, встречающиеся в тренировочной части набора. По полученным значениям метрик можно сделать вывод о том, что выбранные модели сильно переобучаются на фиксированном наборе слов, входящем в тренировочную часть набора HKR. Такой вывод можно сделать благодаря существенному подъему качества распознавания на наборе test1, а также небольшому снижению показателей качества на наборе test2.

По этим результатам видно, что влияние новых почерков на качество распознавания минимально, если еще учитывать тот факт, что все изображения набора имеют похожий стиль. При этом добавление дополнительных обучающих данных с новыми словами позволяет снизить эффект переобучения на фиксированном корпусе и поднять средние значения метрик качества обученной модели на обоих тестовых наборах. Эффект переобучения на словах также может быть связан с тем, что набор данных HKR содержит большое количество изображений с одинаковым текстом: при количестве элементов свыше 45 тысяч, уникальных текстовых единиц всего лишь 2808 (см. табл. 1), некоторые из них незначительно отличаются друг от друга. При этом валидационная часть набора состоит преимущественно из слов, содержащихся в тренировочной части (9133 слова из 9375). Таким образом, обучение только на данных из тренировочной части набора HKR недостаточно оправданно, необходимы дополнительные данные с изображениями новых слов.

По итогам анализа результатов можно сделать вывод о том, что метод генерации дополнительного синтетического набора данных с помощью рукописных шрифтов не сильно уступает другим методам в плане повышения качества обучения моделей распознавания. В среднем, он позволяет повысить точность распознавания (accuracy) обученной модели на 6%, по сравнению с обучением модели на наборе без использования дополнительно сгенерированных данных. Этот метод имеет одно несомненное преимущество перед другими рассмотренными методами – он не требует обучения тяжеловесных моделей и затачивания под конкретный набор данных.

Код с реализацией экспериментов, а также реализация генератора рукописного текста находятся в общем доступе: https://github.com/NastyBoget/hrtr (эксперименты), https://github.com/NastyBoget/HandwritingGeneration (генератор). Кроме того, в открытый доступ выложены наборы данных, сгенерированные тремя способами для наборов данных HKR и Cyrillic Handwriting Dataset:

- synthetic_hkr (https://huggingface.co/datasets/nastyboget/synthetic_hkr);
- stackmix_hkr (https://huggingface.co/datasets/nastyboget/stackmix_hkr);
- gan_hkr (https://huggingface.co/datasets/nastyboget/gan_hkr);
- synthetic_cyrillic (https://huggingface.co/datasets/nastyboget/synthetic_cyrillic);
- stackmix_cyrillic (https://huggingface.co/datasets/nastyboget/stackmix_cyrillic);
- gan_cyrillic (https://huggingface.co/datasets/nastyboget/gan_cyrillic).

6. Заключение

В данной статье описывается разработанный авторами метод создания кириллических рукописных шрифтов, а также метод генерации изображений рукописного текста на их основе. Проведено сравнение предложенного метода с существующими решениями путем

обучения двух моделей распознавания рукописного текста. Модели обучались на двух различных общедоступных наборах данных, дополненных изображениями, сгенерированными различными способами. По результатам исследований можно сделать вывод о том, что разработанный метод сравним по результатам повышения качества распознавания моделей с другими методами. Он позволяет повысить точность распознавания обученной модели в среднем на 6%, при этом не требует обучения дополнительных моделей.

Код разработанного метода, а также проведённых экспериментов выложен в открытый доступ. Сгенерированные в рамках исследования наборы данных доступны для скачивания на платформе huggingface.co.

Дальнейшие исследования могут быть направлены на повышение качества распознавания выбранных моделей: можно убрать верхний порог количества эпох обучения и более тонко скорректировать гиперпараметры, сгенерировать большее количество обучающих данных, а также одновременно использовать дополнительные наборы данных, сгенерированные тремя вышеописанными методами.

Список литературы / References

[1] Abdallah A., Hamada M., Nurseitov D. Attention-Based Fully Gated CNN-BGRU for Russian Handwritten Text. Journal of Imaging, vol. 6, issue 12, 2020, article no. 141, 23 p.

[2] Shonenkov A., Karachev D. et al. StackMix and Blot Augmentations for Handwritten Text Recognition. arXiv preprint arXiv:2108.11667, 2021, 10 p.

[3] Fogel S., Averbuch-Elor H. et al. ScrabbleGAN: Semi-Supervised Varying Length Handwritten Text Generation. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4324-4333.

[4] Cyrillic Handwriting Dataset. Available at: https://www.kaggle.com/datasets/constantinwerner/cyrillic-handwriting-dataset, accessed 02.05.2023.

[5] Nurseitov D., Bostanbekov K. et al. Handwritten Kazakh and Russian (HKR) database for text recognition. Multimedia Tools and Applications, vol. 80, issue 21-23, 2021, pp. 33075 - 33097.

[6] Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов. Доклады Академии наук СССР, том 163, ном. 4, 1965, стр. 845-848 / Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, vol. 10, no. 8, 1966, pp. 707-710.

[7] Krishnan P., Jawahar C.V. Generating Synthetic Data for Text Recognition. arXiv preprint arXiv:1608.04224, 2016, 5p.

[8] Goodfellow I., Pouget-Abadie J. et al. Generative adversarial networks. Communications of the ACM, vol. 63, issue 11, 2020, pp. 139-144.

[9] Kang L., Riba P. et al. GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images. Lecture Notes in Computer Science, vol. 12368, 2020, pp. 273-289.

[10] Krishnan P., Kovvuri R. et al. TextStyleBrush: Transfer of Text Aesthetics from a Single Example. IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access), 2023, 12 p.

[11] Calligraphr. Available at: https://www.calligraphr.com, accessed 02.05.2023.

[12] База сегментированных рукописных символов / Segmented Handwriting Character Base. Available at: https://drive.google.com/folderview?id=0B0EQUc5HmgcGS0l2RDlKenlpNnc&usp=sharing, accessed 02.05.2023 (in Russian).

[13] Sueiras J. Continuous Offline Handwriting Recognition using Deep Learning Models. arXiv preprint arXiv:2112.13328, 2021, 210 p.

[14] Kass D. Vats E. AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks. Lecture Notes in Computer Science, vol. 13237, 2022, pp. 507-522.

[15] Sutskever I., Vinyals O., Le Q.V. Sequence to sequence learning with neural networks. In Proc. of the 27th International Conference on Neural Information Processing Systems, vol. 2, 2014, pp. 3104-3112.

[16] He K., Zhang X. et al. Deep Residual Learning for Image Recognition. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[17] Hochreiter S., Long Short-term Memory, Neural computation, vol. 9, issue. 8, 1997, pp. 1735-1780.

[18] Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473, 2014, 15 p.

[19] Marti U.-V., Bunke H. The IAM-database: an English sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition, vol. 5, issue 1, 2002, Pp. 39–46.
[20] Timakin V., Afanasyev M. A modern approach to the end-to-end bilingual handwriting text recognition on the example of Russian school notebooks. Available at: https://github.com/t0efL/end2end-HKR-research, accessed 02.05.2023.
[21] Liu Z., Mao H. et al. A Convnet for the 2020s. In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.
[22] Graves A., Fernández S. et al. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proc. of the 23rd International Conference on Machine Learning, 2006, pp. 369-376.
[23] Vaswani A., Shazeer N. et al. Attention is all you need. In Proc. of the 31st Conference on Neural Information Processing System, 2017, pp. 5998-6008.
[24] Википедия / Wikipedia. Available at: https://ru.wikipedia.org, accessed 02.05.2023 (in Russian).

Информация об авторах / Information about authors

Анастасия Олеговна БОГАТЕНКОВА является студентом магистратуры кафедры мистемного программирования. Научные интересы: распознавание структуры документов, цифровая обработка изображений, создание искусственных данных.

Anastasiya Olegovna BOGATENKOVA is a master's student of the Department of System Programming. Research interests: document layout analysis, digital image processing, generation of artificial data.

Оксана Владимировна БЕЛЯЕВА является аспирантом, стажером-исследователем. Научные интересы: анализ шаблонов документов, анализ структуры документов, цифровая обработка изображений, нейросетевая обработка данных, распознавание образов компьютерного зрения, распознавание лиц.

Oksana Vladimirovna BELYAEVA is a PhD student, researcher. Research interests: document layout analysis, document structure analysis, digital image processing, neural network data processing, image pattern recognition, face recognition.

Андрей Игоревич ПЕРМИНОВ является аспирантом, стажером-исследователем. Его научные интересы включают цифровую обработку сигналов, нейросетевую обработку данных, создание искусственных данных.

Andrey Igorevich PERMINOV is a PhD student, researcher. His research interests include digital signal processing, neural network data processing, generation of artificial data.