УДК 004.422+004.9

ПОСТ-КОРРЕКЦИЯ СЛАБОЙ РАСШИФРОВКИ БОЛЬШИМИ ЯЗЫКОВЫМИ МОДЕЛЯМИ В ИТЕРАЦИОННОМ ПРОЦЕССЕ РАСПОЗНАВАНИЯ РУКОПИСЕЙ

В. П. Зыков[1] [0009-0007-8935-9288], Л. М. Местецкий[2] [0000-0001-6387-167X]

[1, 2]Московский государственный университет имени М. В. Ломоносова, г. Москва, Россия

[2]НИУ Высшая школа экономики, г. Москва, Россия

[1]zykovvp@my.msu.ru, [2]mestlm@mail.ru

Аннотация

Рассмотрена задача ускорения построения точной редакторской разметки рукописных архивных текстов в рамках инкрементного цикла обучения на основе слабой расшифровки. В отличие от ранее опубликованных результатов, основное внимание уделено интеграции автоматической посткоррекции слабой расшифровки с помощью больших языковых моделей (Large Language Models, LLM). Предложен и реализован протокол применения LLM на уровне строк в режиме обучения на нескольких примерах с тщательно сконструированными промптами и контролем формата вывода (сохранение дореформенной орфографии, защита имен и числительных, запрет на изменение структуры строк). Эксперименты проведены на корпусе дневников А. В. Сухово-Кобылина. В качестве базовой модели распознавания использована строчная версия модели Vertical Attention Network. Результаты показали, что LLM-коррекция на примере сервиса ChatGPT-4o заметно улучшает читабельность слабой разметки и существенно снижает процент ошибок в словах (в нашем опыте – порядка −12 процентных пунктов), при этом не внося ухудшения в проценте ошибок в буквах. Другой исследуемый сервис – DeepSeek-R1 – показал менее стабильное поведение. Рассмотрены практические настройки промптов, ограничения (контекстные лимиты, риск «галлюцинаций») и даны рекомендации по безопасной интеграции LLM-коррекции в итерационный пайплайн разметки с целью сокращения трудозатрат эксперта-асессора и ускорения оцифровки исторических архивов.

***Ключевые слова**: распознавание рукописного текста, слабая разметка, Vertical Attention Network* (*VAN*)*, большие языковые модели* (*LLM*)*, посткоррекция, итерационное дообучение.*

ВВЕДЕНИЕ

Автоматическое распознавание рукописных текстов (handwritten text recognition, HTR) остается важной задачей цифровой гуманитаристики: расшифровка архивных дневников открывает доступ к уникальным историческим материалам и позволяет применять инструменты поиска и анализа текста в гуманитарных исследованиях. Одновременно исторические архивы имеют специфические трудности: встречаются дореформенная орфография, частые зачеркивания, многоязычные вставки, значительные различия в качестве сканирования и нелинейная кривизна строк, что делает прямое применение стандартных HTR-конвейеров малоэффективным без адаптации к конкретному корпусу.

В настоящей работе рассмотрен корпус рукописных дневников А. В. Сухово-Кобылина – важного исторического деятеля XIX в. [1]. Этот архив содержит порядка 10000 страниц, но экспертная разметка есть лишь для 92 страниц (≈2876 полностью размеченных строк), что существенно ограничивает возможности обучения современных нейросетевых моделей. Разметка рукописных текстов XIX в. – это высокоспециализированная, крайне трудоемкая и время затратная работа, доступная только ограниченному кругу исследователей-гуманитариев, обладающих соответствующей филологической подготовкой. Поэтому мы хотим максимально снизить трудозатраты таких высококвалифицированных экспертов на разметку архивных рукописей.

Для подготовки входных данных мы использовали процедуры сегментации и нормализации строк, разработанные в наших предыдущих работах и реализованные в программном комплексе (см. [2, 3]). Эти алгоритмы позволяют выделять базовые линии, корректировать кривизну строк и приводить их к формату, пригодному для строчных моделей распознавания. В качестве базовой архитектуры распознавания бралась строчная версия архитектуры Vertical Attention Network (VAN) [4], т. к. она продемонстрировала конкурентоспособные результаты на открытых наборах

данных при распознавании параграфов и строк на дневнике Ф. П. Литке [5] и использовалась в предыдущей работе [3]. В выборе строчной версии VAN для данного архива ключевую роль сыграли две практические причины: возможность обучаться на отдельно размеченных строках, что важно при частично размеченных страницах; а также устойчивость к сложным нелинейным структурам строк, характерным для нашего корпуса.

Под термином «**слабая расшифровка (разметка)**» мы понимаем автоматическую расшифровку, полученную моделью распознавания на неразмеченных данных и содержащую значительное число ошибок (как на уровне символов, так и на уровне слов). Эксперт-асессор исправляет ошибки в слабой расшифровке, получая таким образом новые высококачественные обучающие примеры.

Инкрементный (итерационный) подход наращивания размеченной выборки, подробно описанный в [3], состоит в цикле:

- обучение модели на имеющейся размеченной выборке;
- применение модели к неразмеченным строкам и получение слабой разметки;
- экспертная корректировка слабой разметки (получение точной редакторской разметки);
- дообучение модели на расширенной выборке и переход к новой итерации.

Такой подход позволяет снизить трудозатраты эксперта-асессора по разметке рукописного текста, т. к. ему нужно лишь исправить ошибки в слабой расшифровке, а не размечать текст с нуля.

В настоящей работе основное внимание уделено интеграции автоматической посткоррекции слабой расшифровки с помощью больших языковых моделей (Large Language Models, LLM) в описанный итерационный цикл. Идея применения LLM в режиме обучения на нескольких примерах (few-shot learning) опирается на подход, описанный в [6], где показано, что LLM могут решать новые языковые задачи, получая только несколько примеров в промпте, без дополнительного дообучения. Мы покажем, что LLM в таком режиме при аккуратно сконструированных промптах и строгих правилах вывода (сохранение дореформенной орфографии, защита имен

и числительных, запрет на изменение структуры строк) способны устранять языковые и морфологические артефакты автоматических транскрипций и повышать читабельность слабой разметки. Описан также протокол применения LLM на уровне строк, проведено сравнение нескольких сервисов (ChatGPT-4o и DeepSeek-R1) и показано, что при корректной настройке промптов LLM может существенно снизить процент ошибок в буквах и уменьшить объем ручных правок, требуемых от эксперта (см. табл. 2).

ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

Область оффлайн-распознавания рукописного текста (HTR) эволюционировала от классических статистических и рекуррентных схем к современным архитектурам с механизмами внимания и трансформерам. Значительный вклад в стандартизацию исследований внесли общедоступные корпуса и соревнования (например, IAM и READ), которые служат основой для сопоставления методов и оценки прогресса [7, 8]. Традиционно в прикладных решениях HTR выделяют два подхода. Первый – это строчный (line-level): страницы предварительно сегментируют на строки одной моделью или алгоритмом, и каждая строка распознается другой моделью. Второй – это страничный (page-level), когда модель принимает на вход большие области текста и генерирует последовательность символов без явной построчной разметки. Классические строчные схемы часто строят на комбинации сверточных блоков и рекуррентных нейронных сетей (recurrent neural networks, RNN) [9], обучение которых происходит через CTC (connectionist temporal classification) – это метод обучения и соответствующая функция потерь, позволяющие обучать модели на последовательных данных без явной разметки по позициям символов [10]. Современные страничные архитектуры (SPAN, OrigamiNet, TrOCR, VAN и др.) включают различные способы формирования 2D-представлений и применения трансформеров с предобученными компонентами [4, 11–13].

Для исторических архивов выбор подхода определяется практическими ограничениями: страничные модели облегчают требования к разметке и лучше учитывают контекст, но предъявляют более строгие требования к форматной стабильности изображений и объему размеченных данных; строчные решения,

напротив, оказываются более гибкими при частично размеченных страницах и сложной геометрии строк, что сделало их предпочтительными в ряде прикладных проектов [4, 7, 8]. Практический опыт проектов по цифровизации (включая Digital Peter [14]) подтверждает, что для исторических данных необходима комбинация надежной сегментации, адаптированных архитектур распознавания и продуманной организационной процедуры разметки.

Ключевым предварительным этапом любого HTR-пайплайна остаются сегментация и нормализация строк: ошибки на этом шаге часто «ломают» последующую обработку, поэтому прикладные системы включают как автоматические алгоритмы детекции и выпрямления строк, так и инструменты ручной корректировки сегментации и аннотаций [2, 8]. При ограниченной экспертной разметке («малые данные») классические подходы по сбору больших размеченных корпусов оказываются нереализуемыми, что стимулирует развитие методов слабой разметки, полу- и самообучения, а также практик итеративного (инкрементного) наращивания размеченной выборки: автоматическая расшифровка применяется к новым данным, затем эксперт правит полученную слабую разметку и модель дообучается на расширенной выборке [3, 14].

Недавний рост качества языковых моделей открывает дополнительное перспективное направление: использование LLM для постобработки автоматических расшифровок. LLM в режиме обучения на нескольких примерах могут исправлять языковые и морфологические артефакты, восстанавливать корректные словоформы и делать выводы о вероятных восстановленных фрагментах текста, что особенно полезно, когда ошибки имеют лингвистический, а не оптический характер. При аккуратной инженерии промптов и ограничениях формата вывода LLM способны сохранять историко-филологические особенности (например, дореформенную орфографию) и тем самым сокращать долю ручной правки со стороны эксперта-асессора [6].

Таким образом, опираясь на наработки по надежной сегментации и организационным процедурам разметки [2, 3], а также на современные достижения в области строчных и страничных архитектур [4, 11–13], целесообразно исследовать практическую интеграцию LLM-коррекции слабых расшифровок в итерационный

пайплайн. Это направление сочетает преимущества автоматического предобработанного вывода компьютерного зрения и языковой постобработки, что потенциально может позволить существенно снизить трудозатраты экспертов и ускорить оцифровку исторических архивов.

ПОСТАНОВКА ЗАДАЧИ

В нашем распоряжении имеется архив рукописных дневников А. В. Сухово-Кобылина, содержащий порядка 10000 сканов страниц. Для части архива доступна экспертная разметка: 92 страницы, что соответствует примерно 2876 полностью размеченным строкам (в текстовых файлах, по одному файлу разметки на страницу). Остальные страницы представлены в виде изображений (рис. 1) и требуют автоматической расшифровки.

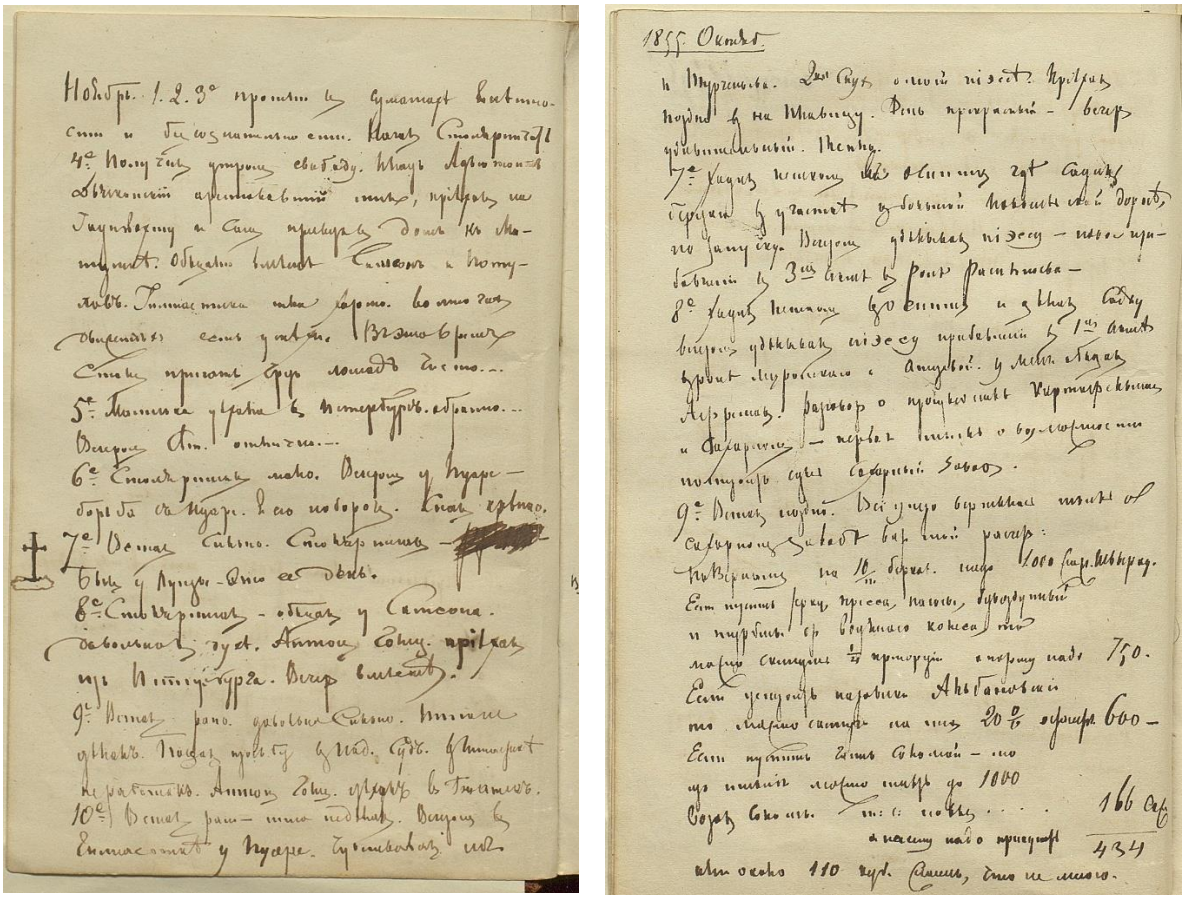

Рис. 1. Примеры страниц дневника А. В. Сухово-Кобылина.

Структура и качество исходных данных характеризуются следующими особенностями, которые определяют практические ограничения задачи:

- сканы различаются по разрешению и качеству, содержат артефакты сканирования;
- тексты написаны дореформенной орфографией XIX в.; присутствуют зачеркивания и вставки на других языках (например на французском);
- строки на странице часто имеют нелинейную (изогнутую) форму; часть строк может быть частично размечена или вовсе не читаема; в случаях явной неразборчивости эксперт отмечает фрагменты специальными маркерами на такие фрагменты и строки мы считаем неразмеченными и исключаем из обучения;
- для подготовки строчной подачи используются алгоритмы сегментации и нормализации строк, описанные в [2, 3]; для распознавания в качестве базовой архитектуры используется строчная версия VAN [4].

Необходимо разработать метод и программную процедуру для автоматической расшифровки неразмеченных страниц с учетом следующих требований.

- Метод должен работать в инкрементном режиме: поддерживать цикл «обучение -> применение -> получение слабой разметки -> экспертная корректировка -> дообучение». Детали организации изложены в [3].
- Метод должен поддерживать интеграцию автоматической LLM-коррекции слабой расшифровки перед этапом экспертной правки (промпт-настройки, сохранение дореформенной орфографии и ограничение на изменение структуры строк). Цель – снизить объем ручных правок, требуемых от эксперта.
- Сохранение историко-филологической корректности: метод не должен автоматически «нормализовывать» дореформенную орфографию без явного указания.

В качестве основного критерия качества распознавания мы будем использовать CER (character error rate) – процент ошибок в буквах. Он рассчитывается следующим образом:

$$\mathrm{CER}(y_{\mathrm{true}}, y_{\mathrm{pred}}) = \frac{\rho_{\mathrm{Lev}}(y_{\mathrm{true}}, y_{\mathrm{pred}})}{\mathrm{length}(y_{\mathrm{true}})} \cdot 100\%,$$

где $\rho_{\mathrm{Lev}}(y_{\mathrm{true}}, y_{\mathrm{pred}})$ – расстояние Левенштейна между правильной экспертной разметкой (y_{true}) и расшифровкой модели (y_{pred}), при этом все строки, на которых подсчитывается указанный критерий, считаются одним цельным текстом.

В качестве вспомогательного критерия качества будем использовать WER (word error rate) – процент ошибок в словах:

$$\mathrm{WER}(y_{\mathrm{true}}, y_{\mathrm{pred}}) = \frac{\tilde{\rho}_{\mathrm{Lev}}(y_{\mathrm{true}}, y_{\mathrm{pred}})}{\mathrm{word\_length}(y_{\mathrm{true}})} \cdot 100\%,$$

в этой метрике подсчет расстояния Левенштейна ($\tilde{\rho}_{\mathrm{Lev}}$) и длины строки ($\mathrm{word\_length}$) происходит на уровне слов, а не отдельных символов.

Необходимо построить и протестировать алгоритм/пайплайн, который при указанных входах и ограничениях минимизирует CER и WER на отложенной тестовой выборке, а также демонстрирует экономию экспертного времени и количества правок при использовании LLM-коррекции.

ПРЕДЛАГАЕМЫЙ ПОДХОД К РАСПОЗНАВАНИЮ

Архитектура сети

Для распознавания рукописного текста нами была использована модель Vertical Attention Network (VAN), описанная в [4]. Эта модель была выбрана, потому что показывает конкурентоспособные результаты на открытых датасетах, хорошо себя зарекомендовала при распознавании дневника Ф. П. Литке [5] и уже использовалась в предыдущей нашей работе [3].

Эта модель имеет две разновидности: страничную и строчную. Страничная модель принимает на вход изображение всей страницы рукописного текста целиком, а строчная работает на уровне отдельных строк.

Строчная архитектура представлена полностью сверточной нейронной сетью, состоящей из кодировщика, содержащего 10 блоков с 3 свертками в каждом

(рис. 2), и сверточного декодировщика, который переводит внутреннее представление модели в набор вероятностей. Архитектура строчной модели схематично представлена на рис. 3.

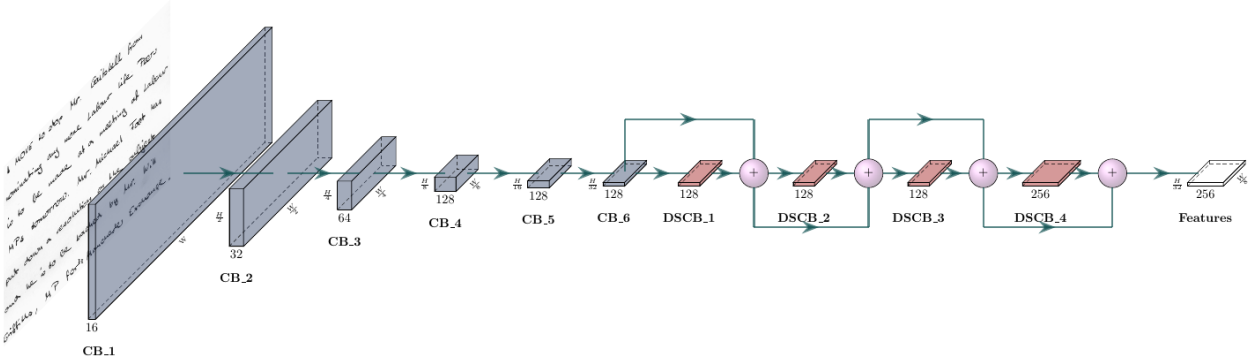

Рис. 2. Кодировщик VAN (одинаковый для страничной и строчной архитектур).

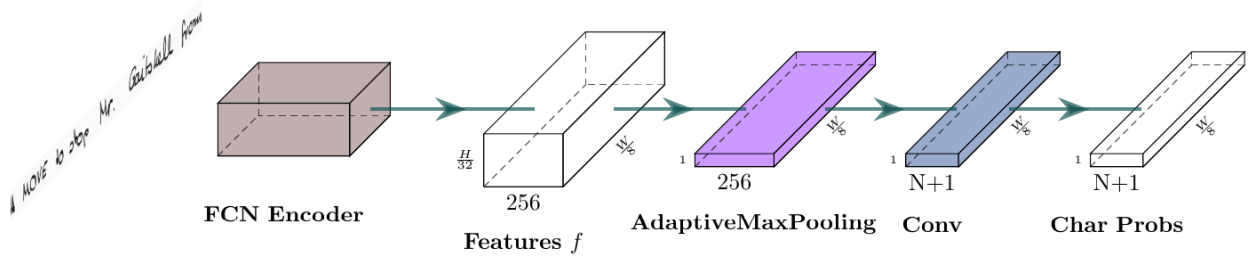

Рис. 3. Строчная архитектура VAN.

Страничная архитектура имеет такой же кодировщик, как и строчная. У страничного варианта модели, в отличие от строчного, есть модуль вертикального внимания (vertical attention), который содержит несколько полносвязных слоев и позволяет рекуррентно собирать представления строк на странице, которые затем приводятся к матрицам вероятностей декодировщиком, содержащим LSTM-слой и сверточный слой 1 x 1. Страничная архитектура изображена на рис. 4. Достоинство страничной модели заключается в том, что она не требует предварительной сегментации строк на странице, но при этом предполагает, что строки текста являются линейными и ровными.

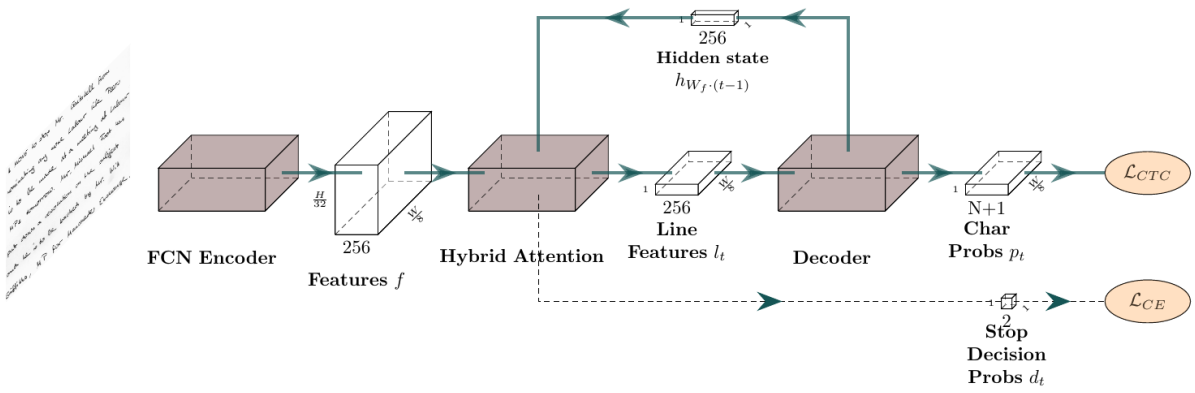

Рис. 4. Страничная архитектура VAN.

Обе модели выдают вероятности символов в строках; расшифровка выполняется жадным декодированием по выходам CTC, а обучение – с использованием функции потерь CTC (Connectionist Temporal Classification) [10]. Таким образом, при обучении решается задача оптимизации

$$\hat{\theta} = \mathrm{argmin}_{\theta} \mathcal{L}_{\mathrm{CTC}}\left(\mathrm{y}, \hat{\mathrm{y}}(\theta)\right),$$

где y – правильная последовательность символов, $\hat{y}(\theta)$ – предсказанная последовательность модели с параметрами θ.

В настоящей работе мы использовали строчный вариант модели VAN. Применительно к рассматриваемой задаче строчная модель имеет несколько преимуществ.

Во-первых, некоторые строки на рукописной странице не удается разобрать и разметить даже эксперту (или разметка этих строк для него просто не представляет научного интереса), но при этом остальные строки на той же странице хорошо читаются. В этом случае непонятно, как использовать такую не до конца размеченную страницу для обучения страничной модели. Строчная же модель такой проблемы не имеет, так как использует для обучения отдельные размеченные строки.

Во-вторых, страничная модель не всегда справляется с проблемой сегментации строк в том случае, когда они имеют сложную криволинейную структуру. Строчная модель может в этом случае игнорировать плохо сегментированные строки на изображении страницы.

В-третьих, с точки зрения эксперта-асессора страница является слишком большим фрагментом текста, эксперт в процессе получения подстрочного перевода оперирует понятиями строк и отдельных слов.

Изначально параметры кодировщика будем инициализировать предобученными весами на англоязычном датасете IAM [7].

Сегментация и нормализация строк

Для используемой нами строчной модели необходима предварительная сегментация изображений страниц на отдельные строки. Эта процедура выполняется с помощью программы «Подстрочник». Описание программы и используемого в ней метода представлены в [2, 3].

Алгоритм сегментации включает следующие шаги.

- Изображение страницы переводится в полутоновое (серое) изображение.
- Для каждой строки изображения (на уровне пикселей) вычисляется сумма яркостей пикселей в этой строке. Получаем функцию, зависящую от ординаты. Эта функция имеет форму синусоиды, минимумы которой соответствуют позициям текстовых строк.
- Страницу можно разделить на сегменты вертикальными линиями и для каждого сегмента выполнить данную операцию, чтобы учесть нелинейность строк.
- Полученные базовые линии строк на разных сегментах объединяются в виде ломаных линий, описывающих положение строк на изображении.
- Далее от базовых линий отступают на некоторое количество пикселей вверх и вниз, чтобы выделить ограничивающие прямоугольники для каждой текстовой строки.
- После этого выполняется нормализация строк – ломаные линии выпрямляются с помощью сдвига значений в столбцах изображения, порождая выпрямленные строки для подачи в строчную модель.

На рис. 5 показаны пример исходного изображения страницы и результат выделения базовых линий строк с использованием программы «Подстрочник».

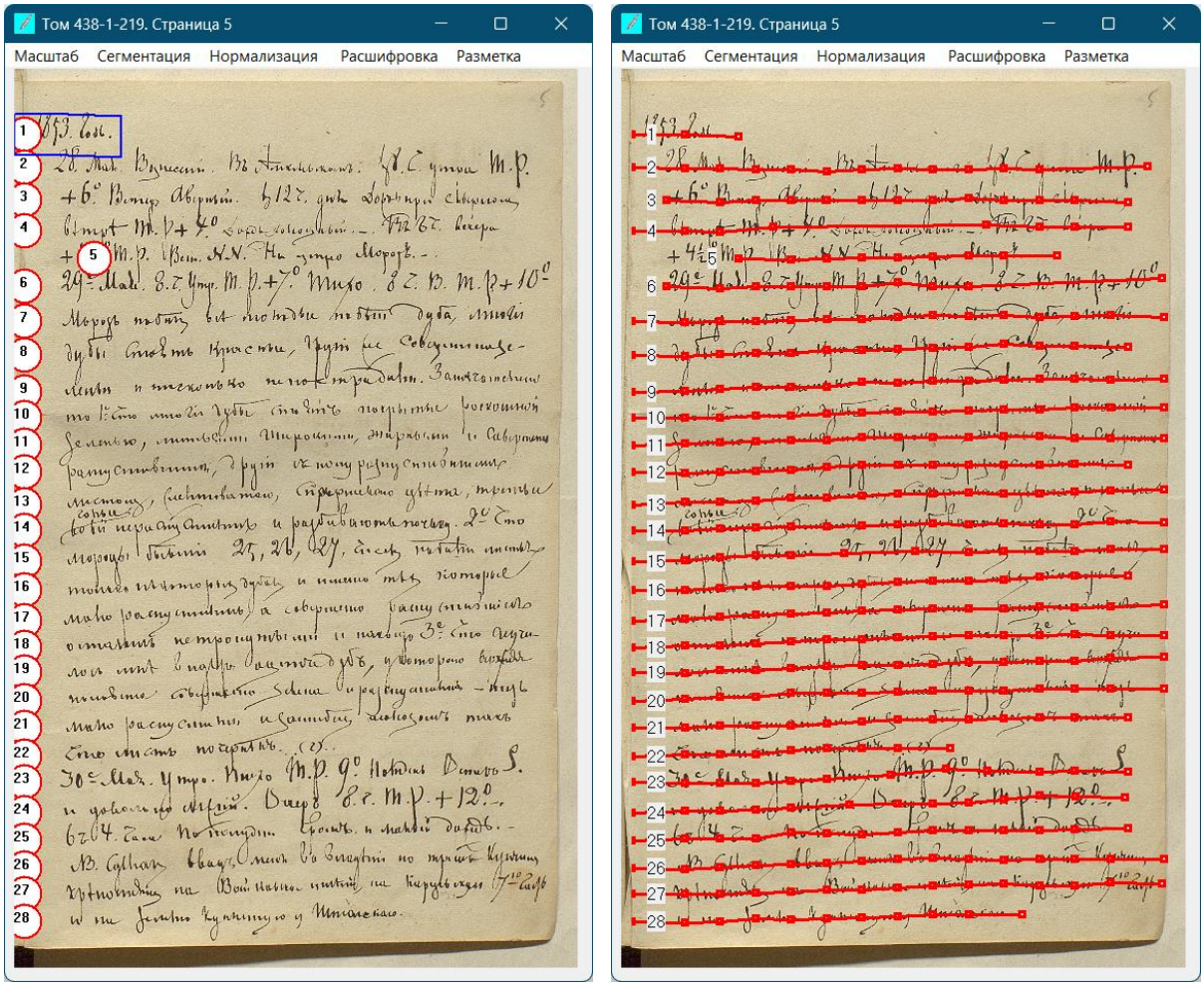

Рис. 5. Исходное изображение страницы, поданное в программу (слева), и изображение с выделенными базовыми линиями строк (справа).

После выделения линий строк выполняется нормализация, в ходе которой строки выпрямляются. На рис. 6 показаны исходная строка и ее нормализованный вид. После этого обработанные строки можно подавать в строчную модель.

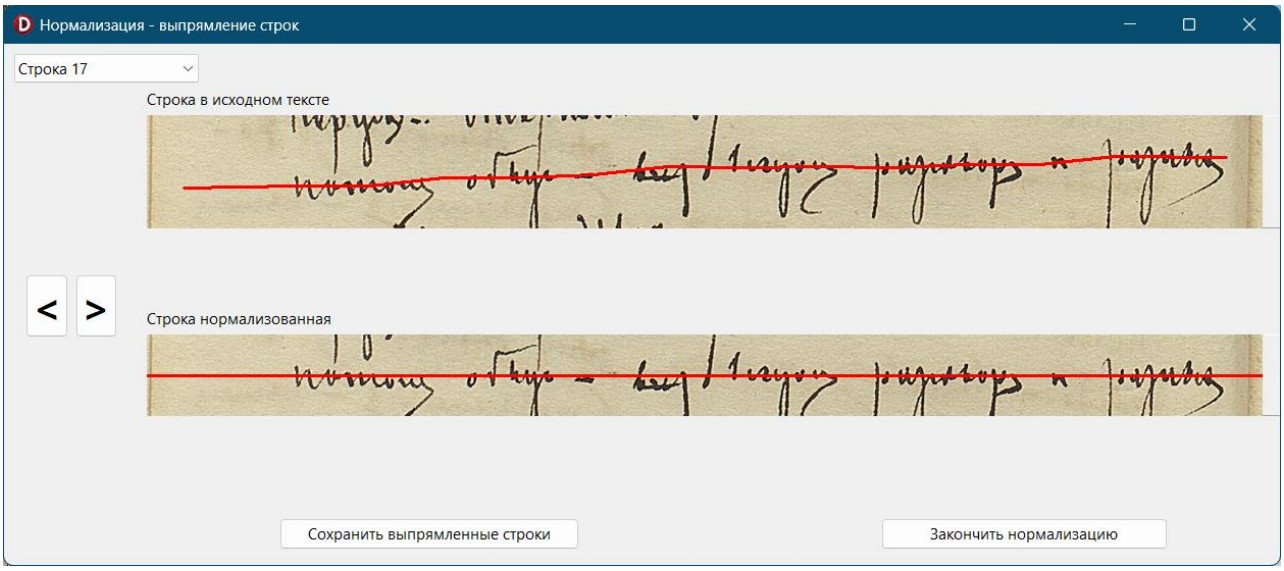

Рис. 6. Нормализация строки: исходная и выпрямленная строка.

Итерационный процесс наращивания обучающей выборки и дообучения модели

Разметка рукописных текстов на русском языке XIX в. является сложной задачей, которую может выполнить только небольшой круг экспертов. Вследствие этого изначально мы имеем довольно небольшое число полностью размеченных строк для обучения нейронной сети.

Предполагается, что увеличить качество распознавания можно путем наращивания обучающей выборки.

Чтобы облегчить эксперту работу по разметке (для увеличения обучающей выборки), мы применили обученную модель на новых, еще неразмеченных рукописных текстах. Полученную после распознавания расшифровку мы называем **слабой разметкой** (или **слабой расшифровкой**). Слабая разметка содержит довольно большое число ошибок. Далее эксперту нужно лишь исправить эти ошибки, чтобы получить редакторскую разметку. Это сделать существенно проще, чем с нуля разметить рукописный текст.

После получения новой порции размеченных данных происходит обучение модели на увеличенной выборке. Это, по предположению, должно увеличить качество распознавания обученного алгоритма. Далее улучшенный алгоритм

можно опять применить на новых данных, после чего эксперту нужно будет исправить меньшее число ошибок по сравнению с предыдущей итерацией. Таким образом, получаем итерационный процесс наращивания обучающей выборки и дообучения модели с увеличением качества распознавания на каждой такой итерации. Этот процесс схематично показан на рис. 7.

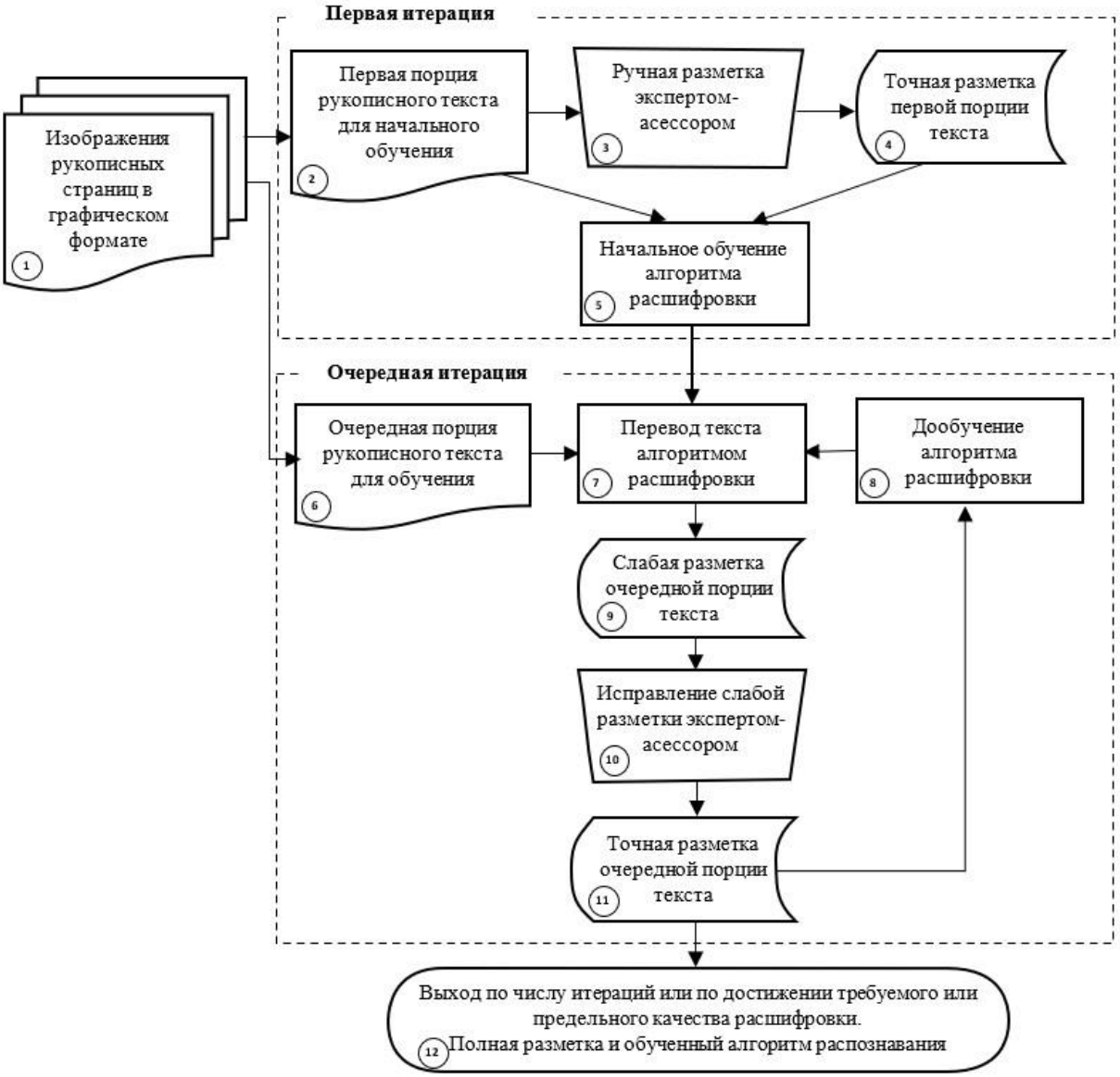

Рис. 7. Схема итерационного процесса распознавания.

Кроме того, работу эксперта упрощает программа «Подстрочник» с графическим интерфейсом. Она позволяет выбрать интересующую эксперта страницу дневника (рис. 8), сделать автоматическую сегментацию строк (рис. 5), исправить

полученную сегментацию в случае обнаружения в ней ошибок, нормализовать строки (рис. 6), применить заранее обученную нейронную сеть для получения автоматической расшифровки модели и отредактировать слабую расшифровку, создав тем самым размеченную для обучения строку (рис. 9).

Более подробное описание программы «Подстрочник», данного итерационного процесса и его экспериментальной корректности дано в [3].

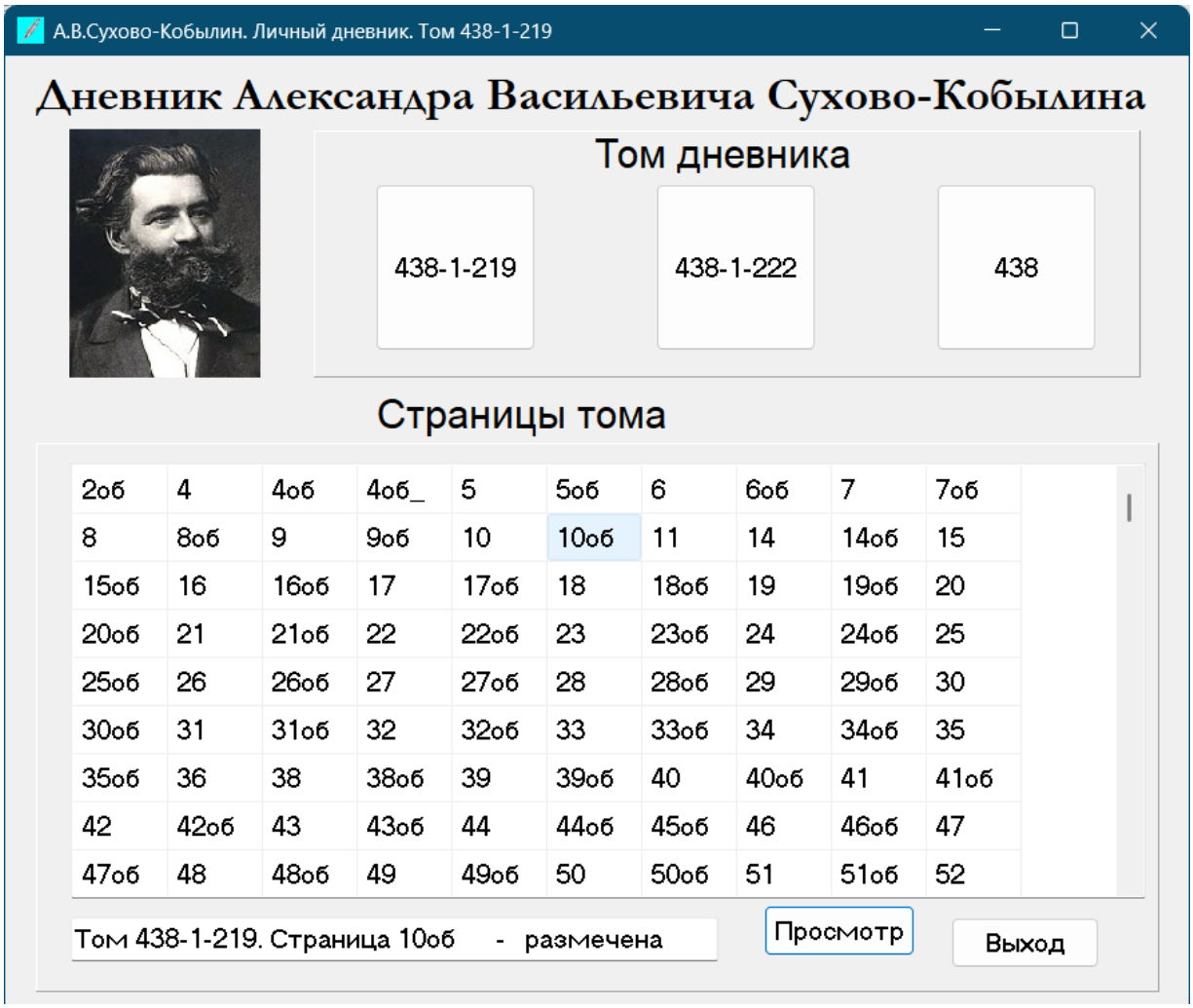

Рис. 8. Программа «Подстрочник»: выбор страницы.

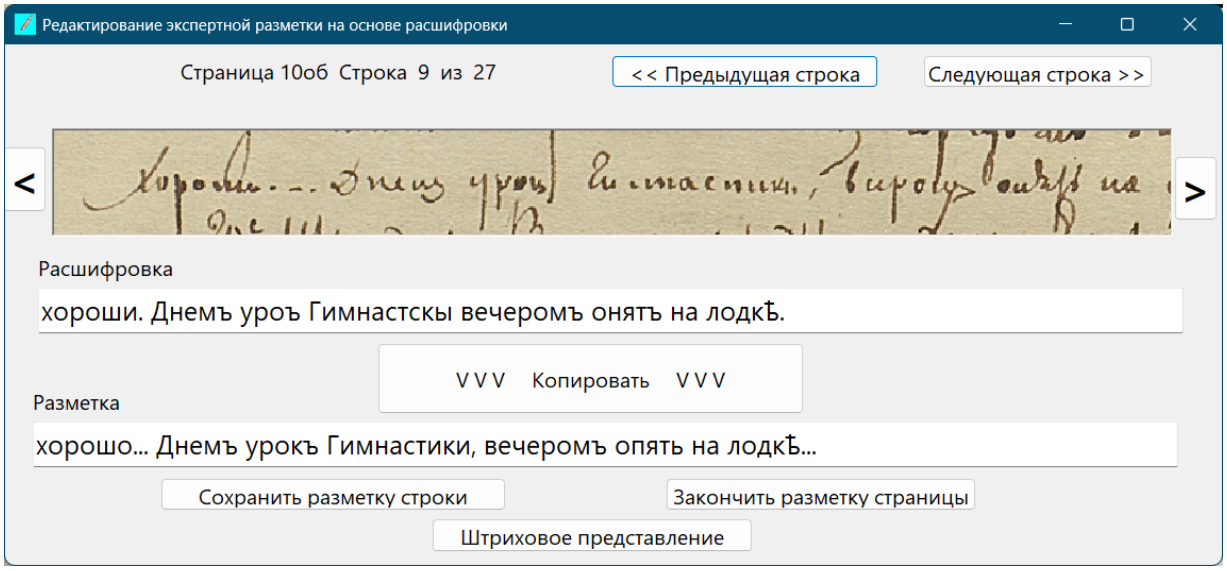

Рис. 9. Программа «Подстрочник»: редактирование расшифровки модели.

ЭКСПЕРИМЕНТЫ

Исходное обучение на архиве А. В. Сухово-Кобылина

В этом эксперименте происходит обучение модели на данных дневника А. В. Сухово-Кобылина. Тренировочная выборка составлена из 1599 строк, валидационная – из 119 строк, тестовая – из 74 строк. Остальные 1084 строки (из указанных в постановке задачи 2876 строк) были не размечены в момент проведения эксперимента.

Для ускорения сходимости модель инициализируется предобученными весами. Для инициализации весов кодировщика использовались веса, полученные при обучении на датасете IAM, а декодировщик обучался с нуля, поскольку алфавит у этих датасетов отличается (в разных датасетах используются разные множества символов).

Применялись аугментации, предложенные в статье про VAN [4]. Был использован batch_size = 128. Остальные гиперпараметры обучения остались такими же, как в [4]. Модель обучалась в течение 8152 эпох, что на видеокарте NVIDIA A100 80GB заняло 1 сутки. Лучшее качество на валидации наблюдалось на 7802-й эпохе.

Оценки качества на эпохе (7802), лучшей с точки зрения CER на валидации, представлены в табл. 1.

Табл. 1. Полученные метрики при исходном обучении

Выборка	CER (%)	WER (%)
Train	3.52	10.80
Valid.	17.74	53.73
Test	15.93	50.64

Примеры работы модели на тестовой выборке показаны на рис. 10. Видно, что даже при текущем уровне ошибок расшифрованный текст оказывается вполне читаемым и по нему можно понять смысл написанного.

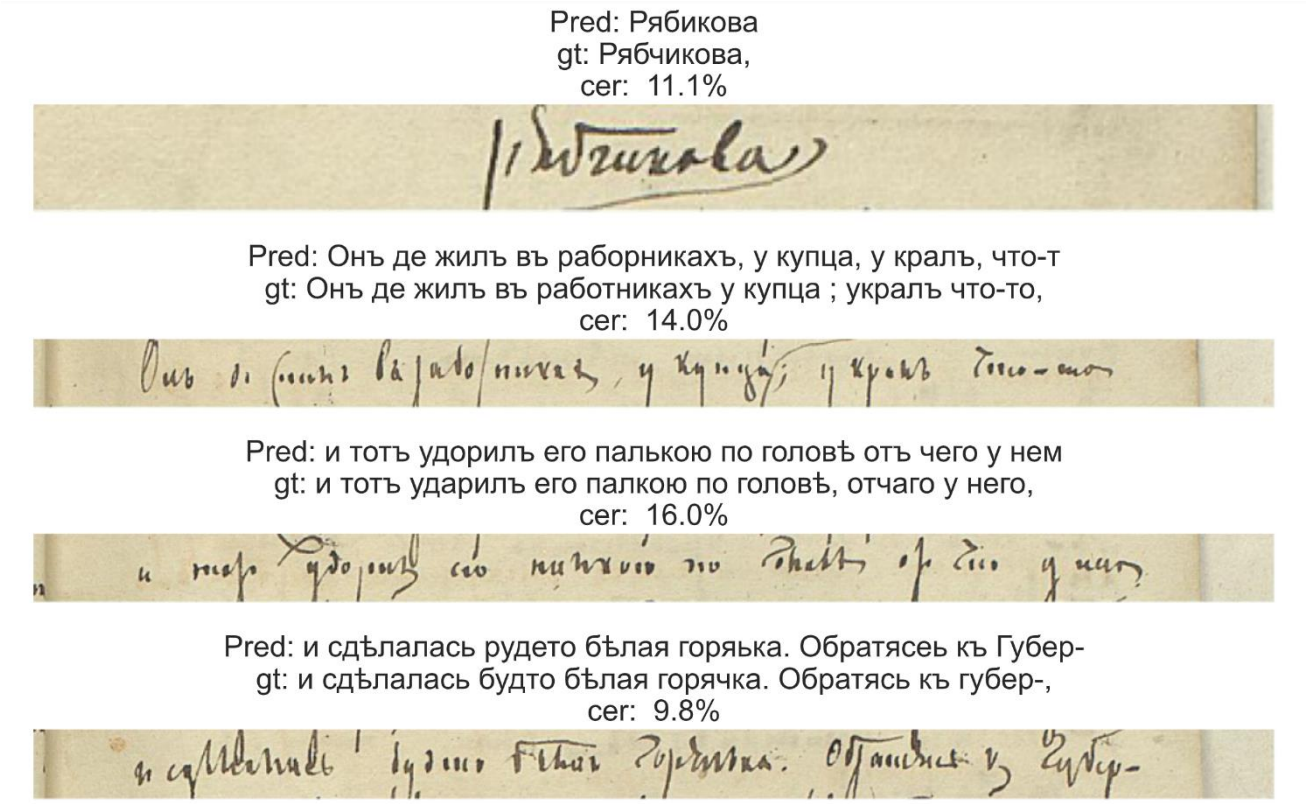

Рис. 10. Примеры работы модели на тестовых данных: Pred – предсказание модели, gt – разметка эксперта.

Однако на тестовых данных ошибки все-таки оказываются довольно большими: CER = 15.93%, WER = 50.64%. Число этих ошибок может уменьшить их исправление с помощью больших языковых моделей, о чем речь пойдет далее.

Исправление ошибок с помощью больших языковых моделей

Цель проведенного эксперимента – проверить гипотезу о том, что большие языковые модели способны повысить качество автоматических расшифровок строк, получаемых строчной моделью VAN, и тем самым снизить объем ручных правок, требующихся от эксперт-асессора в рамках итерационного цикла.

Исправление выполнялось на уровне отдельных строк (коррекция целых строк), чтобы не изменять количество строк на странице, это важно для интеграции в рабочий интерфейс и последующего дообучения. В эксперименте использовались две LLM: ChatGPT-4o и DeepSeek-R1. В качестве подхода применялся режим обучения на нескольких примерах: в системный промпт включались примеры пар «исходная строка -> исправление» из валидационной выборки (119 примеров). Для ChatGPT дополнительно были поданы 250 примеров из тренировочной выборки; у DeepSeek-R1 технически не удалось разместить столь большой контекст примеров. Ключевые инструкции промпта были такие: сохранять исходную последовательность слов и структуру строк, не выполнять нежелательную нормализацию дореформенной орфографии, не изменять имена собственные и числительные, исправлять только явные опечатки и ошибки распознавания. Пример системного промпта:

> *«Твоя задача – корректировать входной текст, исправляя в нем ошибки.*
> *Это текст, распознанный моделью компьютерного зрения с рукописей. Рукописи написаны на русском языке 19-го века (также иногда присутствуют другие языки).*
> *Цель – получить максимально близкий к рукописи вариант расшифровки.*
> *Модель допускает ошибки в распознавании символов.*
> *Исправляй только самые явные и понятные места. Если фрагмент текста сложно разобрать, то сохраняй его в том же виде, в котором и получил.*
> *Сохраняй имена собственные и числительные как есть. Сохраняй исходную последовательность слов.*
> *Тебе будут подаваться строки, распознанные моделью компьютерного зрения, к которым в конце добавлены символы ``->''. Тебе нужно вернуть исправленную строку.*

Примеры:
1856. Мартъ -> 1856. Мартъ
3-е. Въ часовъ пріѣхалъ въ Канугу. Дядѣ принялъ меня -> 3е. Въ 11 часовъ пріѣхалъ въ Калугу. Дядя принялъ меня
лучше. Стотрѣли иланхъ моего завода – онъ далъ мнѣ -> лучше. Смотрѣли планы моего Завода – онъ далъ мнѣ
Отдалъ перепиывать піэссу въ Печать. -> Отдалъ переписывать піэссу въ печать.
<Остальные примеры ...>»

Был использован режим сохранения истории: при последовательной правке строк предыдущие исправления и ответы ассистента оставлялись в контексте, чтобы LLM получала дополнительный локальный контекст и лучше понимала смысл соседних строк.

На рис. 11 и 12 представлены примеры исправлений с помощью указанных больших языковых моделей.

В табл. 2 приведены показатели CER и WER на тестовых данных после исправления большими языковыми моделями, а также изменения метрик по сравнению с отсутствием исправлений (предыдущий эксперимент). Из таблицы видно, что ChatGPT оставляет CER примерно таким же, но при этом значительно снижает процент ошибок в словах ($\Delta\mathrm{WER} = -12.27\%$). DeepSeek же работает в этом случае хуже, он часто выдает больше одной строки, поэтому он увеличивает CER, что является неприемлемым; WER он уменьшает не так сильно, как ChatGPT. Из-за ограничения на размер промпта DeepSeek-R1 принял только 119 примеров исправлений из валидационной выборки, тогда как для ChatGPT-4o в системный промпт дополнительно было включено ещё 250 примеров из тренировочной выборки. Таким образом, использование ChatGPT позволило значительно снизить ошибки модели компьютерного зрения (строчной VAN), а использование DeepSeek не принесло положительных результатов.

```
Model:   Самояло ва презнательные А.В. Сухово.– Ко- (cer = 15.38%)
ChatGPT: Самойлова признательные А. В. Сухово-Ко- (cer = 2.56%)
Human:   Самойлова признательные А.В. Сухово-Ко-

Model:   28-е. Сборъ на Выксу. Проицаные съ Самсономъ. (cer = 11.63%)
ChatGPT: 28е. Сборъ на Выксу. Прощанье съ Самсономъ. (cer = 4.65%)
Human:   28е. Сборы на Выксу. Прощаніе съ Самсономъ.

Model:   продалъ d пудъ патоки но 180 к. сереб. и п.р. ереб. (cer = 27.59%)
ChatGPT: продалъ 2 пуда патоки по 180 к. сереб. и 1 р. сереб. (cer = 22.41%)
Human:   Продалъ 1 т. пудов патоки по 1.80 к. сер. и 1 т. р. сереб.

Model:   вшесъ Рертеру въ упламу за аппараты Къ сахар (cer = 14.89%)
ChatGPT: внёсъ Рейтеру въ уплату за аппараты къ сахару (cer = 12.77%)
Human:   внесъ Ферстеру въ уплату за аппараты къ сахаръ-
```

Рис. 11. Примеры работы ChatGPT: исходные строки модели VAN (Model), исправленные ChatGPT строки (ChatGPT) и разметка эксперта (Human).

```
Model:    Самояло ва презнательные А.В. Сухово.– Ко- (cer = 15.38%)
DeepSeek: Самоялову. Признательные А. В. Сухово. – Ко- (cer = 23.08%)
Human:    Самойлова признательные А.В. Сухово-Ко-

Model:    28-е. Сборъ на Выксу. Проицаные съ Самсономъ. (cer = 11.63%)
DeepSeek: 28-е. Сборъ на Выксу. Прощаніе съ Самсономъ. (cer = 4.65%)
Human:    28е. Сборы на Выксу. Прощаніе съ Самсономъ.

Model:    продалъ d пудъ патоки но 180 к. сереб. и п.р. ереб. (cer = 27.59%)
DeepSeek: продалъ 10 пудъ патоки по 180 к. сереб. и 5 р. сереб. (cer = 22.41%)
Human:    Продалъ 1 т. пудов патоки по 1.80 к. сер. и 1 т. р. сереб.

Model:    вшесъ Рертеру въ упламу за аппараты Къ сахар (cer = 14.89%)
DeepSeek: въ счетъ Ретерту въ уплату за аппараты. Къ сахар- (cer = 25.53%)
Human:    внесъ Ферстеру въ уплату за аппараты къ сахаръ-
```

Рис. 12. Примеры работы DeepSeek: исходные строки модели VAN (Model), исправленные DeepSeek строки (DeepSeek) и разметка эксперта (Human).

Табл. 2. Сравнение метрик CER и WER после корректировки расшифровки тестовых данных с помощью ChatGPT-4o и DeepSeek-R1

Модель	CER (%)	ΔCER (%)	WER (%)	ΔWER (%)
ChatGPT-4o	15.87	–0.06	38.37	–12.27
DeepSeek-R1	17.39	+1.46	42.08	–8.56

Вероятно, LLM не специализированы на рукописи русского языка XIX в. и вряд ли в явном виде обучались на больших корпусах дореформенной орфографии; тем не менее при соответствующем промптинге ChatGPT в эксперименте продемонстрировал способность корректно обрабатывать такие формы и в большинстве случаев сохранять дореформенную орфографию (то есть не нормализовывать текст к современному правописанию). Это указывает на то, что аккуратная инженерия промпта и ограничение правок (жесткие правила) позволяют эффективно применять LLM для посткоррекции исторических транскриптов.

Полученные результаты показали, что LLM-коррекция может быть полезным встроенным этапом итерационного пайплайна: автоматическая посткоррекция повышает читабельность слабой разметки и уменьшает долю очевидных ошибок, которые эксперту требуется исправить вручную. В частности, значительное снижение WER у ChatGPT подразумевает сокращение числа словесных правок и, как следствие, потенциальную экономию времени эксперта при создании точной редакторской разметки. При этом важно сохранять контроль поведения LLM (четкие промпты, лимиты выдачи, постфильтрация), поскольку некоторые модели могут изменять структуру строки или «предполагать» недостающие фрагменты – поведение, неприемлемое для автоматического включения в обучающий пул без ручной проверки.

В текущей реализации наблюдаются следующие ограничения: размер контекстного промпта влияет на качество – возможность подать больше примеров (как у ChatGPT) дает преимущество; некоторые LLM склонны к перестроению строки (изменению длины/формата), что плохо сочетается со строчным пайплай-

ном; риск «галлюцинаций» и нежелательной нормализации требует строгих правил промпта и постфильтров. Исходя из этого, для практической интеграции LLM в итерационный цикл следует: применять LLM-коррекцию на уровне строк с жестким контролем формата вывода; использовать исторический контекст для повышения согласованности исправлений; добавлять этап автоматической валидации результатов LLM (проверка длины, допустимых символов, сохранение имен и чисел) перед передачей строк эксперту.

В целом эксперимент подтвердил, что LLM-коррекция может быть эффективно использована в пайплайне инкрементной разметки: при аккуратной инженерии промптов и фильтрации выводов она повышает читабельность слабой разметки и убирает часть ошибок, которые в противном случае потребовали бы ручной правки.

ЗАКЛЮЧЕНИЕ

Предложен и экспериментально исследован практический подход к ускорению построения точной редакторской разметки исторических рукописей в рамках инкрементного (итерационного) цикла: основная новация – это интеграция автоматической посткоррекции получаемой «слабой» расшифровки с помощью больших языковых моделей. В качестве базового распознавателя использовалась строчная версия Vertical Attention Network (VAN), обученная на размеченных строках корпуса дневников А. В. Сухово-Кобылина; исходные метрики на тестовой выборке составили CER ≈ 15.93% и WER ≈ 50.64%.

Экспериментальная оценка показала, что LLM-коррекция при аккуратной инженерии промптов и контроле формата вывода действительно уменьшает долю ошибок на уровне слов. Наиболее стабильный выигрыш продемонстрировал ChatGPT-4o: WER на тесте снизился примерно на 12.3 процентных пункта (до 38.37%) при практически неизменном CER (≈15.87%). Другой протестированный сервис (DeepSeek-R1) показал менее приемлемое поведение: снижение WER сопровождалось ростом CER и неоднократным «перестроением» строк, что делает автоматическое включение таких исправлений в обучающий пул рискованным без ручной валидации.

Практическая значимость полученных результатов состоит в следующем. Автоматическая LLM-коррекция повышает читабельность слабой разметки и уменьшает объем явных ошибок в словах, которые эксперт-асессор должен исправлять вручную; это потенциально приводит к заметной экономии времени эксперта в процессе итерационного наращивания размеченной выборки. Для безопасной и воспроизводимой интеграции LLM в пайплайн необходимо соблюдать ряд мер предосторожности: жесткую спецификацию промптов (с сохранением дореформенной орфографии, защитой имен и числительных и запретом на изменение структуры строк), использование исторического контекста для повышения согласованности, автоматическая поствалидация вывода (проверки длины / набора символов / сохранения ключевых токенов).

Ограничения проведенной работы также очевидны и важны для интерпретации результатов: зависимость качества от выбора и размера примеров в промпте (context length), стоимость и вопросы приватности при использовании коммерческих LLM, риск «галлюцинаций» и перестроения строки в неконтролируемых моделях, а также то, что эксперименты выполнены на одном корпусе с конкретными свойствами (дореформенная орфография, разное качество сканов, зачеркивания). Эти факторы накладывают практические ограничения на немедленную массовую автоматизацию.

На основе проведенной работы сформулированы практические рекомендации для внедрения LLM-коррекции в пайплайны распознавания исторических рукописей:

- применять LLM-коррекцию на уровне строк с жесткими ограничениями на формат вывода;
- конструировать промпты с явными примерами (в рамках подхода обучения на нескольких примерах) и контролировать историю исправлений для согласованности;
- внедрять автоматическую поствалидацию и фильтрацию результатов LLM перед передачей эксперту;
- учитывать затраты и приватность при выборе LLM.

В будущем мы планируем расширить и углубить исследование в нескольких ключевых направлениях. В частности, предстоит исследовать стратегии селекции данных для разметки. Одно из направлений – это оценка показателей ансамблевой неопределенности (по аналогии с [15]) как одного из сигналов приоритизации строк. Предварительный анализ показал, что простая реализация неопределенности может указывать на «трудные» строки, однако ее прямое применение к отбору обучающих примеров не гарантирует устойчивого улучшения. Поэтому необходима более тщательная проработка (калибровка ансамбля, предфильтрация по качеству сегментации, комбинирование сигналов неопределенности и представительности). Кроме того, планируются изучение альтернативных мер неопределенности, методов агрегации ансамблей, тестирование различных LLM и промпт-стратегий, а также масштабирование подхода на другие корпуса исторических рукописей.

Таким образом, работа показала практическую ценность использования LLM для посткоррекции автоматических расшифровок в рамках итерационного пайплайна – это реальный путь к сокращению ручной работы экспертов при оцифровке архивов. Одновременно установлены важные ограничения и направления доработки, реализация которых сделает предложенный подход более надежным, масштабируемым и пригодным для промышленного применения.

Благодарность

Работа поддержана грантом РНФ №22-68-00066 «Культурное наследие России: интеллектуальный анализ и тематическое моделирование корпуса рукописных текстов».

СПИСОК ЛИТЕРАТУРЫ

1. *Пенская Е.Н., Купцова О.Н.* Невидимая величина. А.В. Сухово-Кобылин: театр, литература, жизнь. М.: Изд. дом ВШЭ, 2024. 472 с.

2. *Местецкий Л.М., Смирнова В.С.* Сегментация строк в изображениях рукописных документов // Материалы Международной конференции по компьютерной графике и зрению (Графикон-2025). Поволжский государственный технологический университет, Йошкар-Ола, Россия, 2025.

3. *Местецкий Л.М., Зыков В.П.* Инкрементная разметка рукописных архивных дневников XIX века // Программные продукты и системы. 2025. Т. 38, № 4. https://doi.org/10.15827/0236-235X.152

4. *Coquenet D., Chatelain C., Paquet T.* End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023. Vol. 45, No. 1. P. 508–524. https://doi.org/10.1109/TPAMI.2022.3144899

5. *Болтунова Е.М., Лаптев А.К.* Распознавание рукописного текста и интеллектуальный анализ: возможности нейронных технологий (на примере работы с «Дневником» Ф.П. Литке) // Имагология и компаративистика. 2025. № 23. С. 358–379. https://doi.org/10.17223/24099554/23/17

6. *Brown T. B., Mann B., Ryder N., Subbiah M. et al.* Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems (NeurIPS). 2020. Vol. 33. P. 1877–1901.

7. *Marti U.-V., Bunke H.* The IAM-database: an English sentence database for offline handwriting recognition // International Journal on Document Analysis and Recognition (IJDAR). 2002. Vol. 5, No. 1. P. 39–46. https://doi.org/10.1007/s100320200071

8. *Sánchez J., Romero V., Toselli A. H., Vidal E.* ICFHR2016 competition on handwritten text recognition on the READ dataset // Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR 2016). 2016. P. 630–635.

9. *Shi B., Bai X., Yao C.* An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39, No. 11. P. 2298–2304. https://doi.org/10.1109/TPAMI.2016.2646371

10. *Graves A., Fernández S., Gomez F., Schmidhuber J.* Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks // Proceedings of the 23rd International Conference on Machine Learning (ICML 2006). 2006. P. 369–376. https://doi.org/10.1145/1143844.1143891

11. *Coquenet D., Chatelain C., Paquet T.* SPAN: A Simple Predict & Align Network for Handwritten Paragraph Recognition // Document Analysis and Recognition – ICDAR 2021. Lecture Notes in Computer Science, Vol. 12823. Springer, 2021. P. 70–84. https://doi.org/10.1007/978-3-030-86334-0_5

12. *Yousef M., Bishop T.E.* OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by Learning to Unfold // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020). 2020. P. 14710–14719. https://doi.org/10.1109/CVPR42600.2020.01472

13. *Li M., Lv T., Chen J., Cui L., Lu Y., Florencio D., Zhang C., Li Z., Wei F.* TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models // Proceedings of the AAAI Conference on Artificial Intelligence. 2023. Vol. 37, No. 12. P. 14216–14224.

14. *Potanin M., Dimitrov D., Shonenkov A., Bataev V., Karachev D., Novopoltsev M., Chertok A.* Digital Peter: New Dataset, Competition and Handwriting Recognition Methods // Proceedings of the 6th International Workshop on Historical Document Imaging and Processing. ACM, 2021. P. 43–48. https://doi.org/10.1145/3476887.3476892

15. *Lakshminarayanan B., Pritzel A., Blundell C.* Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles // Advances in Neural Information Processing Systems (NeurIPS). 2017. Vol. 30. P. 6402–6413.

POST-CORRECTION OF WEAK TRANSCRIPTIONS BY LARGE LANGUAGE MODELS IN THE ITERATIVE PROCESS OF HANDWRITTEN TEXT RECOGNITION

V. P. Zykov[1] [0009-0007-8935-9288], L. M. Mestetskiy[2] [0000-0001-6387-167X]

[1, 2]Lomonosov Moscow State University, Moscow, Russia

[2]HSE University, Moscow, Russia

[1]zykovvp@my.msu.ru, [2]mestlm@mail.ru

Abstract

This paper addresses the problem of accelerating the construction of accurate editorial annotations for handwritten archival texts within an incremental training cycle based on weak transcription. Unlike our previously published results, the present work focuses on integrating automatic post-correction of weak transcriptions using large language models (LLMs). We propose and implement a protocol for applying LLMs at the line level in a few-shot setup with carefully designed prompts and strict output format control (preservation of pre-reform orthography, protection of proper names and numerals, prohibition of structural changes to lines). Experiments are conducted on the corpus of diaries by A.V. Sukhovo-Kobylin. As the base recognition model, we use the line-level variant of the Vertical Attention Network (VAN). Results show that LLM post-correction–exemplified by the ChatGPT-4o service–substantially improves the readability of weak transcriptions and significantly reduces the word error rate (in our experiments by about −12 percentage points), without degrading the character error rate. Another service tested, DeepSeek-R1, demonstrated less stable behavior. We discuss practical prompt engineering, limitations (context length limits, risk of “hallucinations”), and provide recommendations for the safe integration of LLM post-correction into an iterative annotation pipeline to reduce expert annotators’ workload and speed up the digitization of historical archives.

Keywords*: handwritten text recognition, weak markup, Vertical Attention Network (VAN), large language models (LLM), post-correction, iterative retraining.*

REFERENCES

1. *Penskaya E.N., Kuptsova O.N.* (2024) The Invisible Quantity. A.V. Sukhovo-Kobylin: Theater, Literature, Life. Moscow: HSE Publishing House, 2024. 472 p. (In Russ.)

2. *Mestetsky L.M., Smirnova V.S.* Line segmentation in images of handwritten documents // Proceedings of the International Conference on Computer Graphics and Vision (Grafikon-2025). Yoshkar-Ola: Volga State Technological University, 2025. (In Russ.)

3. *Mestetskiy L.M., Zykov V.P.* Incremental markup of 19th-century handwritten ar-chival diaries // Software & Systems. 2025. Vol. 38, No. 4. https://doi.org/10.15827/0236-235X.152. (In Russ.)

4. *Coquenet D., Chatelain C., Paquet T.* End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023. Vol. 45, No. 1. P. 508–524. https://doi.org/10.1109/TPAMI.2022.3144899

5. Boltunova E.M., Laptev A.K. Handwriting recognition and data mining: Possibilities of neural network technologies (based on admiral Fyodor Lutke's diary) // Imagology and Comparative Studies. 2025. No. 23. P. 358–379. https://doi.org/10.17223/24099554/23/17. (In Russ.)

6. *Brown T.B., Mann B., Ryder N., Subbiah M. et al.* Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems (NeurIPS). 2020. Vol. 33. P. 1877–1901.

7. *Marti U.-V., Bunke H.* The IAM-database: an English sentence database for offline handwriting recognition // International Journal on Document Analysis and Recognition (IJDAR). 2002. Vol. 5, No. 1. P. 39–46. https://doi.org/10.1007/s100320200071

8. *Sánchez J., Romero V., Toselli A. H., Vidal E.* ICFHR2016 competition on handwritten text recognition on the READ dataset // Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR 2016). 2016. P. 630–635.

9. *Shi B., Bai X., Yao C.* An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition // IEEE

Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39, No. 11. P. 2298–2304. https://doi.org/10.1109/TPAMI.2016.2646371

10. *Graves A., Fernández S., Gomez F., Schmidhuber J.* Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks // Proceedings of the 23rd International Conference on Machine Learning (ICML 2006). 2006. P. 369–376. https://doi.org/10.1145/1143844.1143891

11. *Coquenet D., Chatelain C., Paquet T.* SPAN: A Simple Predict & Align Network for Handwritten Paragraph Recognition // Document Analysis and Recognition – ICDAR 2021. Lecture Notes in Computer Science, Vol. 12823. Springer, 2021. P. 70–84. https://doi.org/10.1007/978-3-030-86334-0_5

12. *Yousef M., Bishop T.E.* OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by Learning to Unfold // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020). 2020. P. 14710–14719. https://doi.org/10.1109/CVPR42600.2020.01472

13. *Li M., Lv T., Chen J., Cui L., Lu Y., Florencio D., Zhang C., Li Z., Wei F.* TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models // Proceedings of the AAAI Conference on Artificial Intelligence. 2023. Vol. 37, No. 12. P. 14216–14224.

14. *Potanin M., Dimitrov D., Shonenkov A., Bataev V., Karachev D., Novopoltsev M., Chertok A.* Digital Peter: New Dataset, Competition and Handwriting Recognition Methods // Proceedings of the 6th International Workshop on Historical Document Imaging and Processing. ACM, 2021. P. 43–48. https://doi.org/10.1145/3476887.3476892

15. *Lakshminarayanan B., Pritzel A., Blundell C.* Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles // Advances in Neural Information Processing Systems (NeurIPS). 2017. Vol. 30. P. 6402–6413.

СВЕДЕНИЯ ОБ АВТОРАХ

ЗЫКОВ Валерий Павлович – магистрант кафедры «Математические методы прогнозирования» факультета вычислительной математики и кибернетики МГУ имени М.В. Ломоносова. Область научных интересов: машинное обучение, распознавание рукописных текстов, математика.

Valerii Pavlovich ZYKOV – Master's student at the Department “Mathematical Forecasting Methods”, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. Research interests: machine learning, handwriting recognition, mathematics.

email: zykovvp@my.msu.ru

ORCID: 0009-0007-8935-9288

МЕСТЕЦКИЙ Леонид Моисеевич – доктор технических наук, академик РАЕН, профессор кафедры математических методов прогнозирования МГУ. Научные интересы: вычислительная геометрия, анализ и распознавание изображений.

Leonid Moiseevich MESTETSKIY – Doctor of Engineering Sciences, Academician of the Russian Academy of Natural Sciences, Professor at the Department of Mathematical Forecasting Methods at Moscow State University. His research interests include computational geometry and image analysis and recognition.

email: mestlm@mail.ru

ORCID: 0000-0001-6387-167X

Материал поступил в редакцию 1 ноября 2025 года