

## АВТОМАТИЗАЦИЯ СТРУКТУРНОЙ РАЗМЕТКИ ИСТОРИЧЕСКИХ РУКОПИСЕЙ: ИНТЕГРАЦИЯ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ И ИНТЕРАКТИВНОГО РЕДАКТИРОВАНИЯ

Бабкин Антон Кириллович<sup>1</sup>, Ушанкова Мария Юрьевна<sup>2</sup>

<sup>1</sup>Студент;

Государственный университет «Дубна»;

141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: anton.epub@gmail.com.

<sup>2</sup>Старший преподаватель;

Государственный университет «Дубна»;

141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: ushankova.m.ju@uni-dubna.ru.

*В работе представлена информационная система для автоматизации структурной разметки архивных рукописей. Решена проблема утраты визуального контекста документа путем привязки текстовой расшифровки к координатам на изображении. Реализована клиент-серверная веб-архитектура с возможностью локального развертывания. Интеграция нейросетевых моделей YOLOv9 и TrOCR позволила автоматизировать сегментацию строк и распознавание текста. Разработан алгоритм сохранения разметки при геометрических преобразованиях сканов. Система обеспечивает формирование качественных обучающих выборок и ускоряет работу архивистов.*

**Ключевые слова:** архивные документы, структурная разметка, машинное обучение, распознавание рукописного текста, цифровизация.

### Для цитирования:

Бабкин А. К., Ушанкова М. Ю. Автоматизация структурной разметки исторических рукописей: интеграция методов машинного обучения и интерактивного редактирования // Системный анализ в науке и образовании. 2026. № 2. С. 1-9. EDN: ISBHLX. URL: <https://sanse.ru/index.php/sanse/article/view/713>.

## AUTOMATION OF STRUCTURAL MARKUP OF HISTORICAL MANUSCRIPTS: INTEGRATION OF MACHINE LEARNING METHODS AND INTERACTIVE EDITING

Babkin Anton K.<sup>1</sup>, Ushankova Maria Yu.<sup>2</sup>

<sup>1</sup>Student;

Dubna State University;

19 Universitetskaya st., Dubna, Moscow region, 141980, Russia;

e-mail: anton.epub@gmail.com.

<sup>2</sup>Senior teacher;

Dubna State University;

19 Universitetskaya st., Dubna, Moscow region, 141980, Russia;

e-mail: ushankova.m.ju@uni-dubna.ru.

*The paper presents an information system for automating the structural markup of archival manuscripts. The problem of the loss of a document's visual context is solved by linking the text transcription to image coordinates. A client-server web architecture with the capability for local deployment has been implemented. The integration of YOLOv9 and TrOCR neural network models enabled the automation of line segmentation and text recognition. An algorithm for preserving markup during geometric transformations of scans was developed. The system ensures the creation of high-quality training datasets and accelerates the work of archivists.*



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

**Keywords:** archival documents, structural markup, machine learning, handwritten text recognition, digitization.

### **For citation:**

Babkin A. K., Ushankova M. Yu. Automation of structural markup of historical manuscripts: integration of machine learning methods and interactive editing. *System analysis in science and education*, 2026;(2):1-9 (in Russ). EDN: ISBHLX. Available from: <https://sanse.ru/index.php/sanse/article/view/713>.

## **Введение**

В фондах научно-исследовательских институтов, таких как Объединенный институт ядерных исследований (ОИЯИ), сосредоточены значительные объемы уникальных материалов: личные дневники ученых, журналы работы экспериментальных установок и архивные газеты. Цифровизация этих документов необходима не только для обеспечения их сохранности как объектов исторического наследия, но и для возможности последующего автоматизированного анализа содержащейся в них информации.

Однако на практике процесс цифровизации идет медленно из-за отсутствия подходящих инструментов. Специалист вынужден работать в текстовых редакторах (*DOCX, TXT*), оторванных от изображения, или в универсальных платформах, не заточенных под рукописи. Метод «двух окон» требует постоянного переключения внимания, что повышает утомляемость и риск механических ошибок.

Главный недостаток такого подхода заключается в полной утрате связи между расшифровкой и визуальным обликом страницы. Для историков и исследователей сохранение исключительно «плоского» текста ведет к невозможной потере визуального контекста документа: особенностей почерка, зачеркиваний, рисунков, вклеек и расположения заметок на полях, которые зачастую несут не меньше исторической информации, чем само содержание (например, свидетельствуют о физическом или эмоциональном состоянии автора).

С технической же стороны, обычный текстовый файл не хранит информацию о координатах конкретных слов на исходном снимке, поэтому его недостаточно для формирования качественных обучающих выборок для нейросетевых моделей. Таким образом, как для исторического анализа, так и для задач оптического распознавания (*OCR/HTR*) критически важно не только иметь транскрипцию, но и точно привязывать её к конкретной области изображения. Кроме того, раздельное хранение расшифровок и сканов в разных каталогах усложняет управление данными, повышает риск ошибок и затрудняет коллективную работу нескольких исследователей над одним документом.

В связи с этим актуальной задачей является разработка информационной системы, объединяющей в едином интерфейсе инструменты для ручной разметки и современные алгоритмы машинного обучения. Это позволит специалистам быстро выделять области на странице и связывать их с текстовой расшифровкой в одном рабочем пространстве, не переключаясь между разными программами. Такой подход не только ускорит текущую обработку архивов, но и обеспечит формирование качественных наборов данных для обучения нейронных сетей.

## **1. Обзор аналогов**

Для решения задачи структурной разметки и распознавания текста (*OCR/HTR*) существует ряд готовых программных решений. Их можно условно разделить на две категории: универсальные платформы для аннотирования данных и узкоспециализированные системы для работы с документами.

### **1.1 Универсальные платформы**

К первому классу относятся такие популярные инструменты, как *Label Studio* и *CVAT*. Эти платформы предоставляют мощный функционал для разметки данных, однако они ориентированы преимущественно на инженеров по машинному обучению. Их интерфейс оптимизирован для

аннотирования изолированных объектов (например, на машинопечатных бланках или чеках) и не приспособлен для потоковой транскрипции сплошного рукописного текста. При работе с архивными документами, содержащими десятки плотно исписанных строк, процесс разметки и ввода текста в таких системах оказывается слишком долгим, неудобным и вызывает быстрое утомление оператора, что делает их малоэффективными для масштабных задач историков и архивистов.

## 1.2 Специализированные платформы

Вторую категорию составляют специализированные *HTR*-платформы, такие как *Transkribus* [1], *eScriptorium* [2] и *OCR4all* [3].

Индустриальный стандарт *Transkribus* демонстрирует высокое качество распознавания и активно используется мировым научным сообществом [1], однако это проприетарное решение, работающее по модели платной подписки и требующее загрузки данных в облако.

Открытые платформы, разворачиваемые локально, предлагают альтернативный подход. Например, инструмент *OCR4all* предоставляет полноценный конвейер для распознавания исторического текста, однако он сфокусирован преимущественно на работе со старопечатными книгами с четкой структурой, а не на сложных рукописных дневниках. Наиболее близким аналогом *Transkribus* является *eScriptorium* [2], поддерживающий полигональную разметку и экспорт в стандарт *PAGE XML*. Тем не менее, практический опыт использования таких систем выявляет общие недостатки: высокий порог входа, громоздкость развертывания и перегруженный интерфейс, в котором затруднено быстрое ручное редактирование контуров. Кроме того, жесткая привязка этих платформ к встроенным *OCR*-движкам (например, *Kraken* или *Calamari*) существенно усложняет подключение альтернативных, более современных ИИ-архитектур.

## 1.3 Практический опыт и требования крупных гуманитарных проектов

Необходимость создания специализированного инструментария подтверждается анализом рабочих процессов в масштабных проектах по цифровизации исторического наследия.

Опыт научно-исследовательского центра «Прожито» (ЕУСПб) наглядно подтверждает заявленную ранее проблему утраты визуального контекста при простом текстовом наборе [4]. Переход к структурной разметке, жестко связывающей расшифровку с координатами на скане, является для архивистов необходимостью. Однако на практике этот процесс тормозится высоким порогом входа: существующие универсальные платформы разметки слишком сложны для непрофильных специалистов (историков, филологов), а их интерфейсы не приспособлены для потоковой работы с плотными рукописными текстами.

В то же время, успешный опыт проекта *Trove* Национальной библиотеки Австралии [5] доказывает, что ключом к эффективной массовой оцифровке является максимальное упрощение пользовательского интерфейса. Отказавшись от перегруженных инструментов редактирования в пользу предельно простой двухпанельной среды («скан – текстовое поле»), библиотека смогла успешно организовать ручную вычитку сотен миллионов строк старых газет после неточного автоматического распознавания.

Таким образом, анализ выявляет четкий запрос на объединение этих подходов. Возникает потребность в легковесном, локально разворачиваемом решении, которое гарантирует точность полигональной привязки (для сохранения исторического контекста и обучения ИИ), но при этом обладает интуитивным интерфейсом, минимизирующим рутинные действия оператора. Отсутствие на рынке системы, сочетающей сложную пространственную разметку с эргономикой простого текстового редактора в рамках закрытого ИТ-контура, обусловило актуальность данной разработки.

## 2. Обоснование технических решений

Архитектура разработанной информационной системы построена на разделении ответственности между серверной частью, отвечающей за хранение и тяжелые вычисления (инференс нейросетевых моделей), и клиентской частью, предоставляющей инструменты интерактивного редактирования.

## 2.1 Выбор программного стека и модели развертывания

В качестве базовой модели была выбрана веб-ориентированная архитектура. Это решение продиктовано необходимостью обеспечения кроссплатформенности и возможности коллективной работы исследователей над общими архивными фондами без необходимости установки специализированного ПО на рабочие станции пользователей.

Серверная часть реализована на языке *Python* с использованием легковесного фреймворка *Flask*. Выбор *Python* обусловлен его статусом стандарта де-факто в области машинного обучения, что позволило бесшовно интегрировать библиотеки глубокого обучения (*PyTorch*, *HuggingFace*, *Ultralytics*) непосредственно в бэкенд системы.

Архитектура системы следует принципам модульности и разделения ответственности. Такая структура сокращает время на внесение изменений.

Использование *Flask* позволило обеспечить низкий порог входа в проект для других разработчиков.

## 2.2 Работа с данными и безопасность

Для хранения структурированной информации (проекты, ссылки на изображения, полигоны, статусы обработки) используется СУБД *SQLite* с прослойкой *SQLAlchemy (ORM)*. Выбор *SQLite* обусловлен её портативностью: база данных хранится в одном файле вместе с архивом изображений. Это не только упрощает перенос данных, но и избавляет от необходимости администрирования серверной СУБД. Для описания геометрии разметки применена гибридная модель данных: основные сущности хранятся в реляционном виде, а координаты полигонов и распознанный текст — в формате *JSON*. Это обеспечивает гибкость при работе с полигонами произвольной формы с переменным количеством вершин, сохраняя при этом высокую скорость выборки данных.

Система развертывается в локальном информационном контуре организации без взаимодействия с внешними облачными сервисами.

Настройки безопасности системы предусматривают два режима функционирования:

- без аутентификации
- ролевая модель с разделением прав доступа «администратор/пользователь». Во втором случае администраторы обладают полным набором прав, тогда как пользователи ограничены операциями просмотра и аннотирования.

Система поддерживает экспорт результатов разметки в формат *PAGE XML* [6], обеспечивая совместимость с другими *HTR*-платформами, а также в *PDF* для просмотра и распространения готовых расшифровок.

## 2.3 Клиентская часть и графический движок

Ключевой задачей при разработке визуальной части системы было предоставление оператору удобных инструментов для работы с векторной графикой поверх фотографий. Для решения этой задачи была выбрана библиотека *Fabric.js*. В первую очередь она предоставляет мощный высокоуровневый интерфейс (*API*) для создания, манипуляции и группировки полигональных объектов.

Дополнительным преимуществом *Fabric.js*, работающей на базе технологии *HTML5 Canvas*, является использование аппаратного ускорения браузера (*GPU*). Это обеспечивает высокую отзывчивость системы и отсутствие визуальных задержек даже при масштабировании изображений с десятками сложных полигонов, что существенно снижает зрительную утомляемость оператора при длительной работе.

На базе данного движка реализован единый технологический конвейер, разделенный на три специализированных редактора:

### Редактор кадрирования

Предоставляет интерактивные инструменты для исправления выравнивания наклона и обрезки лишних полей документа.

### Редактор сегментации

Обеспечивает разметку полигональных областей текстовых строк, включая автоматическую детекцию ИИ-моделью с последующим ручным уточнением полученных контуров.

### Редактор транскрипции

Редактор транскрипции состоит из двух панелей и имеет два режима работы:

- Визуальный режим: на левой панели отображается исходный документ с контурами полигонов, на правой панели - изображение с наложенным распознанным текстом. Выбор полигона инициирует открытие модального окна с увеличенным фрагментом и полем ввода; нажатие клавиши Enter сохраняет транскрипцию и осуществляет автоматический переход к следующему полигону.
- Режим блокнота: правая панель заменяется на вертикальный список текстовых полей, соответствующих полигонам. Оператор осуществляет последовательный ввод, переключаясь между строками клавишей. Данный режим оптимизирован для переноса текстовых данных из внешних источников или при работе с документами не требующих поэлементного увеличения.

Совместно с предметными специалистами выработаны соглашения об обозначении особых случаев: слабо зачёркнутые, но читаемые фрагменты передаются с использованием форматирования ~текст~; квадратные скобки используются как универсальный маркер комментария – для неразборчивых мест, редакторских пояснений и прочих отступлений.

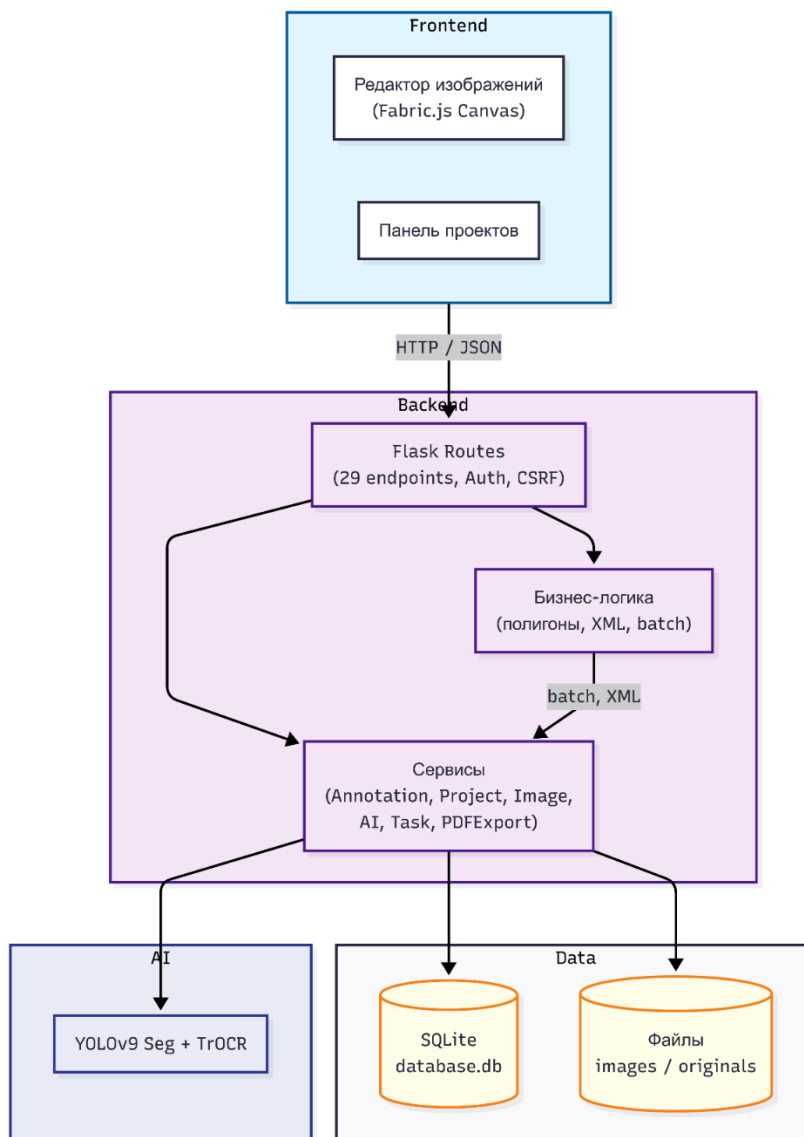


Рис. 1. Архитектура информационной системы

## 2.4 Организация процесса разметки и жизненный цикл документа

В отличие от классических конвейеров машинного обучения, где разметка является лишь подготовительным этапом, в разработанной системе процесс аннотирования представляет собой самостоятельную исследовательскую работу. Для его структурирования реализована статусная модель.

Жизненный цикл каждого изображения описывается строгой последовательностью этапов: «Загружено» → «Обрезано» → «Полигоны готовы» → «Текст распознан» → «Проверено». Переход между статусами осуществляется как автоматически (после обработки алгоритмов ИИ), так и вручную. Присвоение итогового статуса «Проверено» возможно исключительно человеком.

## 3. Описание алгоритмов и моделей

Автоматизированный конвейер обработки архивных документов включает в себя последовательное применение методов компьютерного зрения для детекции структуры страницы и оптического распознавания символов.

### Алгоритм геометрического преобразования и сохранения разметки

При подготовке архивных сканов часто требуется скорректировать наклон страницы или изменить границы кадра. В системе реализован механизм, позволяющий выполнять такие операции без потери созданной разметки.

#### Аффинное преобразование изображения

Поворот и кадрирование выполняются с использованием бикубической интерполяции (*Pillow*), что минимизирует размытие символов при произвольных углах поворота.

#### Алгоритм пересчёта координат полигонов

Данный алгоритм позволяет сохранить созданную разметку, даже если пользователь решит повторно повернуть или кадрировать изображение. Процесс состоит из двух шагов:

1. Привязка к оригиналу: все точки полигонов переводятся из пикселей текущего рабочего экрана в относительные координаты исходного снимка. Это позволяет "закрепить" разметку на оригинальном файле вне зависимости от того, как он повернут или обрезан в текущем редакторе.
2. Проекция на новый кадр: при любом изменении наклона или границ кадра система заново вычисляет положение точек относительно нового холста.

Благодаря такому подходу оператору не нужно перерисовывать полигоны и заново вводить текст после уточнения границ кадра или выравнивания горизонта.

## 4. Интеграция алгоритмов ИИ

Автоматизация ключевых операций реализована посредством интеграции предобученных нейросетевых моделей, выполняемых в фоновом режиме. Данный архитектурный подход позволяет осуществлять вычислительно затратные вычисления без блокировки пользовательского интерфейса, обеспечивая интерактивность при работе с массивами документов.

### 4.1 Модель сегментации строк

Для автоматического поиска строк текста и определения их границ используется нейросетевая архитектура *YOLOv9* [7]. В отличие от классических детекторов, оперирующих прямоугольными рамками (*bounding boxes*), в данной системе применена модификация для сегментации экземпляров (*instance segmentation*).

Выходные маски модели преобразуются в векторные полигоны, точно описывающие естественные изгибы рукописных строк. Результаты автоматической сегментации отображаются в редакторе полигонов, где оператор осуществляет визуальную верификацию и при необходимости

корректирует геометрию контуров интерактивными средствами до перехода к этапу распознавания. К полученным контурам применяется алгоритм геометрического упрощения, что позволяет сократить количество узловых точек полигона без потери точности охвата текста. Это существенно упрощает работу оператора при корректировке формы строки.

## 4.2 Модель распознавания текста (TrOCR)

Для задачи транскрипции (*HTR*) используется архитектура *TrOCR* (*Transformer-based Optical Character Recognition*) [8], объединяющая визуальный трансформер (*ViT*) для кодирования изображения и языковую модель (*RoBERTa*) для генерации текста. Выбор данной модели обусловлен её устойчивостью к вариативности почерка и способностью учитывать контекст соседних символов.

В системе реализован алгоритм подготовки данных для инференса *TrOCR*:

1. Для каждого полигона, детектированного *YOLOv9*, вычисляется описывающий прямоугольник по крайним координатам его вершин.

2. Производится вырезание фрагмента изображения с программным добавлением отступов (*padding*) в 10 пикселей. Это необходимо для сохранения выносных элементов букв (например, «у», «р»), потеря которых при автоматической сегментации критически снижает точность распознавания.

3. Полученное прямоугольное изображение подается на вход *TrOCR*.

Интеграция данных моделей в единый конвейер позволяет достичь высокой степени автоматизации: непосредственно в интерфейс графических редакторов встроены инструменты для запуска алгоритмов ИИ. Оператор может в один клик получить «черновую» разметку всей страницы, после чего незамедлительно переходит к этапу верификации и коррекции данных в том же рабочем пространстве. Такой подход минимизирует рутинные операции и значительно сокращает время подготовки архивных документов.

Система также поддерживает режим пакетной обработки, при котором алгоритмы ИИ применяются ко множеству изображений.

## 5. Оценки качества

На данном этапе исследование было сфокусировано на оценке эргономики инструментов и производительности разработанных алгоритмов. Было проведено очное пользовательское тестирование с замерами времени на выполнение рутинных операций по оцифровке рукописного дневника.

### 5.1 Оценка модуля кадрирования

Тестирование подтвердило высокую интуитивность специализированного инструмента кадрирования: на обработку пакета из пяти изображений у пользователя ушло около 1 минуты.

### 5.2 Оценка модуля сегментации

Сравнение ручной и автоматической разметки строк показало значительное преимущество интеграции машинного обучения. Полностью ручная отрисовка полигонов на одной странице занимала у пользователя от 5 до 10 минут. Использование модели *YOLOv9* позволило автоматизировать этот процесс: после однократной настройки гиперпараметров (порога уверенности, упрощения полигонов и степени слияния областей) разметка одной страницы занимала менее одной минуты. По оценкам пользователей, ручная корректировка автоматически сгенерированных полигонов воспринимается субъективно легче и выполняется значительно быстрее, чем создание разметки с нуля.

### 5.3 Оценка инструментов транскрипции

Тестирование модели *TrOCR* показало, что на сложных исторических рукописях модель демонстрирует низкое качество. В связи с этим упор в системе сделан на эргономику ручного ввода.

В ходе эксперимента сравнивалось время транскрипции в традиционном текстовом редакторе (*MS Word*) и в интерфейсе разработанной системы. Классический набор текста ожидаемо оказался более быстрым. Использование пофрагментного ввода в разработанной системе требует на 20-30% больше времени из-за необходимости навигации между отдельными текстовыми полями.

Инструмент «Блокнот» показал высокую эффективность в сценариях, когда у пользователя уже имеется готовая (ранее набранная в сторонних редакторах) текстовая расшифровка, которую необходимо связать с графическими полигонами. Пользователь вставляет весь массив текста и с помощью горячих клавиш последовательно распределяет его по сегментам.

На основе собранной обратной связи сформулированы задачи по дальнейшей оптимизации интерфейса, включая расширение клавиатурной навигации между сегментами и внедрение интерактивных подсказок для гиперпараметров *YOLOv9*.

## 6. Возможная масштабируемость

Разработанная архитектура системы закладывает прочный фундамент для поэтапного горизонтального и вертикального масштабирования при увеличении объемов обрабатываемых фондов и расширении штата исследователей. Гибкость решения обеспечивается применением слоистой архитектуры: четким разделением на веб-слой маршрутизации, слой бизнес-логики и слой доступа к данным. Такая структура позволяет модифицировать или заменять отдельные компоненты системы без риска нарушения целостности всего приложения.

На этапе локального развертывания и тестирования система использует встроенную СУБД *SQLite*, возможностей которой достаточно для проектов объемом до десятков тысяч изображений. Использование паттерна *Repository* и *ORM*-библиотеки *SQLAlchemy* полностью абстрагирует бизнес-логику от конкретного диалекта SQL. Это означает, что при переходе к промышленной эксплуатации с сотнями тысяч документов миграция на высокопроизводительные реляционные СУБД (например, *PostgreSQL*) потребует лишь изменения конфигурационной строки подключения, без рефакторинга основного кода.

Аналогичным образом спроектирована работа с файловой системой. Благодаря инкапсуляции всех операций с файлами внутри единственного модуля, система готова к переходу на распределенные S3-совместимые хранилища. Такая миграция потребует модификации методов только одного сервиса, в то время как вся высокоуровневая логика обработки изображений, работы с аннотациями и проектами останется нетронутой.

В текущей реализации инференс нейросетевых моделей выполняется в фоновом режиме с использованием асинхронной модели выполнения задач. Однако строгая изоляция модуля *ai\_service* позволяет в перспективе перейти к микросервисной архитектуре (*Model-as-a-Service*). При росте нагрузки модуль искусственного интеллекта может быть вынесен на выделенный *GPU*-кластер, а связь с основным веб-сервером будет осуществляться через распределенные очереди задач.

Для развёртывания используется *Docker*-конфигурация с поддержкой *NVIDIA Container Toolkit*, позволяющая запускать систему на *GPU* без привязки к конкретной версии операционной системы или драйверов хоста.

Таким образом реализуется принцип *data flywheel*: система на начальном этапе функционирует как эргономичный инструмент ручной транскрипции, однако каждая верифицированная страница пополняет размеченный корпус. По мере накопления данных становится возможным дообучение *HTR*-модели на материалах конкретного архива, после чего ручной ввод постепенно замещается постредактированием машинных гипотез, а исправления оператора в свою очередь улучшают модель.

## Заключение

В результате работы создана информационная система, объединяющая инструменты интерактивной разметки и алгоритмы машинного обучения в едином локально развёртываемом рабочем пространстве. Реализованный подход устраняет ключевые проблемы, выявленные при анализе существующих решений: необходимость переключения между несколькими окнами программ, утрату визуального контекста документа и отсутствие координатной привязки транскрипции к изображению. Транскрипции хранятся совместно с изображениями в единой базе данных, что исключает рассинхронизацию прогресса расшифровки и упрощает коллективную работу.

Интеграция модели *YOLOv9* существенно сократила трудозатраты на сегментацию строк, а автоматически сгенерированные полигоны служат удобной основой для верификации оператором, что соответствует парадигме *human-in-the-loop*. Модель *TrOCR* на сложных исторических рукописях демонстрирует ограниченное качество, однако накапливаемый корпус верифицированной разметки создаёт основу для её дообучения на материалах конкретных архивных фондов. Алгоритм пересчёта координат полигонов при повторном кадрировании устранил необходимость перерисовки разметки после уточнения геометрии снимка.

Слоистая архитектура с изолированными модулями хранения, бизнес-логики и работы моделей обеспечивает готовность к масштабированию системы.

## Список источников

1. Transkribus - A Service Platform for Transcription, Recognition and Retrieval of Historical Documents / P. Kahle, S. Colutto, G. Hackl, G. Mühlberger // 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017. Vol. 4. Pp. 19-24. <https://doi.org/10.1109/ICDAR.2017.307>
2. eScriptorium: An Open Source Platform for Historical Document Analysis / B. Kiessling, R. Tissot, P. Stokes, D. S. B. Ezra // 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). IEEE, 2019. Vol. 2. Pp. 19-19. <https://doi.org/10.1109/ICDARW.2019.10032>.
3. OCR4all — An Open-Source Tool Providing a (Semi-)Automatic Evaluation Pipeline for Historical Printings / C. Reul, D. Christ, A. Hartelt [et al.] // Applied Sciences. 2019. Vol. 9, No. 22. Pp. 4853 <https://doi.org/10.3390/app9224853>
4. Булгаков И., Скоринкин Д. Как устроена оцифровка дневников: Михаил Мельниченко о проекте «Прожито» и его перезапуске // Системный Блокъ: Онлайн-журнал. Системный Блокъ, 2018–2026. URL: <https://sysblok.ru/interviews/kak-ustroena-ocifrovka-dnevnikov-mihail-melnichenko-o-proekte-prozhito-i-ego-perezapuske/> (дата публикации: 18.01.2024).
5. Home - Trove // National Library of Australia: [сайт]. URL: <https://trove.nla.gov.au/> (дата обращения: 05.04.2026).
6. Pletschacher S., Antonacopoulos A. The PAGE (Page Analysis and Ground-Truth Elements) Format Framework // 2010 20th International Conference on Pattern Recognition. IEEE, 2010. Pp. 257-260. DOI: 10.1109/ICPR.2010.72.
7. Wang C.Y., Yeh I.H., Liao H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information // arXiv : [electronic archive]. arXiv: 2402.13616. <https://doi.org/10.48550/arXiv.2402.13616>. Submitted on 21 Feb 2024.
8. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models / Minghao Li, Tengchao Lv, Jingye Chen [et al.] // arXiv : [electronic archive]. arXiv:2109.10282. <https://doi.org/10.48550/arXiv.2109.10282>. Submitted on 21 Sep 2021.