DOI: 10.15514/ISPRAS-2024-36(3)-13

Восстановление текстового слоя PDF документов со сложным фоном

[1] *М.В. Загородников,* ORCID: 0009-0003-9076-4863 <*mishazagorodnikov@mail.ru*>
[1,2] *А.А. Михайлов,* ORCID: 0000-0003-4057-4511 <*mikhailov@icc.ru*>

[1] *Институт динамики систем и теории управления им. В.М. Матросова СО РАН, 664033, Россия, г. Иркутск, ул. Лермонтова, д. 134.*
[2] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. В статье рассматривается формат PDF как инструмент для хранения и передачи документов. Особое внимание уделяется проблеме преобразования данных из формата PDF обратно в исходный формат. Актуальность исследования обусловлена широким использованием формата PDF в электронном документообороте современных организаций. Однако, несмотря на удобство использования PDF, извлечение информации из таких документов может быть затруднено из-за особенностей хранения информации в формате и отсутствия эффективных инструментов для обратного преобразования. В работе предлагается решение, основанное на анализе потока вывода текстовой информации формата PDF. Это позволяет автоматически распознавать текст в PDF-документах, даже если в них есть нестандартные шрифты, сложный фон и повреждена кодировка. Исследование представляет интерес для специалистов в области электронного документооборота, а также для разработчиков программного обеспечения, занимающихся созданием инструментов для работы с PDF.

Ключевые слова: кодировка; PDF; документы; CNN; извлечение; текст.

Для цитирования: Загородников М.В., Михайлов А.А. Восстановление текстового слоя PDF документов со сложным фоном. Труды ИСП РАН, том 36, вып. 3, 2024 г., стр. 189–202. DOI: 10.15514/ISPRAS–2024–36(3)–13.

Благодарности: Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

Recovering Text Layer from PDF Documents with Complex Background

[1] *M.V. Zagorodnikov ORCID: 0009-0003-9076-4863<mishazagorodnikov@mail.ru>*
[1,2] *A.A. Mikhailov ORCID: 0000-0003-4057-4511 <mikhailov@icc.ru>*

[1] *Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, 134, Lermontova st., Irkutsk, 664033, Russia.*
[2] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

Abstract. The article considers PDF as a tool for storing and transferring documents. Special attention is paid to the problem of converting data from PDF back to its original format. The relevance of the study is due to the widespread use of PDF in electronic document management of modern organizations. However, despite the convenience of using PDF, extracting information from such documents can be difficult due to the peculiarities of information storage in the format and the lack of effective tools for reverse conversion. The paper proposes a solution based on the analysis of the text information from the output stream of the PDF format. This allows automatic recognition of text in PDF documents, even if they contain non-standard fonts, complex backgrounds, or damaged encoding. The research is of interest to specialists in the field of electronic document management, as well as software developers involved in creating tools for working with PDF.

Keywords: encoding; PDF; documents; CNN; extraction; text.

For citation: Zagorodnikov M.V., Mikhailov A.A. Recovering Text Layer from PDF Documents with Complex Background. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 3, 2024. pp. 189-202 (in Russian). DOI: 10.15514/ISPRAS-2024-36(3)-13.

Acknowledgements. The research was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation (project No. 1023110300006-9).

1. Введение

Формат PDF[1] представляет собой удобный инструмент для хранения и передачи документов. Преобразование документов и текстов в формат PDF не требует значительных усилий. Однако, преобразование данных из формата PDF обратно в исходный формат может представлять сложность из-за особенностей хранения информации в PDF и отсутствия простых и эффективных инструментов для выполнения таких операций. В современных организациях активно используется электронный документооборот, в большинстве случаев с применением формата PDF. Это может привести к проблеме невозможности быстрого извлечения необходимой информации в случае нарушения кодировки шрифтов в документе. В такой ситуации сотруднику, работающему с документами, приходится вручную переписывать весь текст, что может занять много времени при большом объеме документа. Особенно это затруднительно, когда документ содержит сложный фон, компоновку или нестандартные шрифты. В таких случаях даже системы распознавания на основе OCR [1,2] могут оказаться неэффективными.

Мы предлагаем решение, которое поможет избежать проблем с извлечением информации из PDF-документов. Оно основано на анализе потока вывода текстовой информации формата PDF, это позволяет автоматически распознавать текст в PDF-документах, даже если в них есть нестандартные шрифты или сложный фон.

2. Особенности формата PDF

Portable Document Format (PDF) представляет собой стандартизированный формат электронных документов, разработанный корпорацией Adobe.

[1] https://opensource.adobe.com/dc-acrobat-sdk-docs/standards/pdfstandards/pdf/PDF32000_2008.pdf

Преимуществами PDF являются:

- кроссплатформенность;
- простота использования;
- широкое распространение;
- открытость стандарта.

Однако у формата есть и недостатки:

- значительный объем файлов;
- ориентация на отображения информации, а не её хранение;
- сложная структура формата.

Формат PDF содержит информацию о текстовых элементах, включая последовательность символов, их расположение на странице и используемый шрифт [3]. При попытке извлечения текста из PDF-документа считываются коды символов. Если кодировка [4] символов в шрифте документа соответствует кодировке символов на компьютере, то получается корректный текст.

Проблемы при извлечении текста могут возникнуть при повреждении кодировки символов в шрифте, хранящемся в PDF-документе. Повреждение кодировки может произойти по следующим причинам:

- сжатие PDF-документа;
- специфика программ для конвертации документов в PDF;
- неправильный выбор кодировки при создании PDF-документа.

Поврежденная кодировка не влияет на отображение данных в PDF, но создает проблемы при извлечении текста. Эта проблема особенно остро проявляется при наличии сложного фона, где существующие методы, основанные на оптическом распознавании символов (OCR), демонстрируют низкую точность.

3. Обзор аналогов

В работе [5] представлена подробная сравнительная характеристика инструментов, предназначенных для извлечения текста из PDF-документа. Для выявления недостатков и преимуществ различных решений были рассмотрены некоторые из них:

- **Adobe acrobat** [6] – приложение от компании Adobe, предназначенное для работы с PDF-документами.
- **Tesseract 4.0** [7] – программное обеспечение с открытым исходным кодом, которое может быть использовано через командную строку или интегрированно в приложения на языках программирования C++ и Python с помощью API.
- **Free-Online OCR**[2] – бесплатный веб-сервис, предоставляющий возможность извлечения текста из PDF-документа.

В ходе сравнительного анализа было установлено, что все существующие аналоги сталкиваются с трудностями при работе с PDF-документами, имеющими сложный шаблон или фон. В отличие от них, предлагаемый нами подход основан на использовании информации, содержащейся в самом PDF-документе, что позволяет эффективно извлекать текст и преодолевать проблемы, возникающие при обработке данного класса документов.

[2] http://www.free-online-ocr.com/

Табл. 1. Особенности аналогов
Table 1. Features of analogues

	Adobe acrobat	Tesseract 4.0	Free-Online OCR
Поддержка нескольких языков	V	V	X
Открытый исходный код	X	V	X
Возможность работать онлайн	X	X	V
Отсутствие ограничения на размер PDF-документа	V	V	X
Корректное извлечение текста из PDF со сложным шаблоном или фоном	X	X	X

4. Метод восстановления текстового слоя

Для восстановления текстового слоя PDF-документов разработан метод, состоящий из пяти этапов (см. рис. 1):

1. **Извлечение шрифтов.** Шрифты встроенные в PDF-документ содержат информацию о том, как должны отображаться символы. На этом этапе необходимо извлечь эти шрифты из документа.
2. **Извлечение глифов.** Глифы – это визуальные представления символов в шрифте. На этом этапе извлеченные шрифты анализируются для получения изображений глифов в формате PNG.
3. **Распознавание изображений глифов с помощью сверточной нейронной сети.** Сверточные нейронные сети (CNN) особенно эффективны для задач распознавания образов, таких как распознавание глифов. CNN обучения на наборе данных изображений глифов и используется для идентификации глифов в извлеченных изображениях.
4. **Сопоставление глифов с символами в тексте.** После того как глифы идентифицированы, они сопоставляются с соответствующими символами в тексте. Это требует знания соответствия глифов символам в используемой кодировке.
5. **Восстановление текста.** После сопоставления всех глифов с символами текст может быть восстановлен.

4.1 Извлечение шрифтов

На начальном этапе процесса восстановления текста из PDF-документа было проведено извлечение шрифтов. Существуют два типа шрифтов, при использовании которых при копировании текста из PDF-документа могут возникнуть несоответствия между отображаемым и оригинальным текстом. Первый случай несоответствия проиллюстрирован на рис. 2.

В случае некорректной работы кодировки шрифта, определяющей выбор глифа для отображения символа в PDF-файле, можно использовать словарь символов. Словарь позволяет получить корректное отображение символов, основываясь на их текстовом представлении. Ситуация, подобная описанной, может возникнуть при работе с CID-шрифтами. Пример подобной ситуации представлен на рис. 3.

В шрифтах, используемых в PDF-документах, может отсутствовать отображение CID (Character IDentifier) в код символа. В таких случаях при извлечении текста из PDF символы CID будут интерпретированы как символы Unicode с соответствующими кодами. Например, символ CID, ссылающийся на глиф под индексом 1, будет интерпретирован как символ Unicode с кодом 1, а символ CID, ссылающийся на глиф под индексом 32, будет интерпретирован как пробел. В некоторых PDF-документах может встречаться ситуация,

когда один шрифт фактически разделен на несколько шрифтов с одним и тем же именем, но содержащих разные глифы. В таких случаях для их корректного отображения в имя шрифта приходится добавлять незначащую строку.

Рис. 1. Схема метода
Fig. 1. Scheme of the method

Рис. 2. Неправильная кодировка
Fig. 2. Incorrect encoding

Рис. 3. Отсутствие кодировки
Fig. 3. Lack of encoding

4.2 Извлечение глифов

На текущем этапе работы из извлеченных шрифтов были получены все содержащиеся в них глифы. Полученные глифы были сохранены в формате PNG с именами, идентичными именам глифов в шрифте. Если имя глифа было корректным и могло быть преобразовано в символ юникода, например «one», то именем соответствующего PNG-файла становился сам символ, в данном случае «1». Пример PNG-файлов с глифами и их именами, полученными из шрифта с некорректной кодировкой, представлен на рис. 4.

Рис. 4. Глифы из шрифта с неправильной кодировкой
Fig. 4. Glyphs from a font with incorrect encoding

В ряде случаев можно столкнуться со шрифтом, в котором отсутствует кодировка. В таком случае глифы шрифта должны иметь имена. В качестве альтернативы имена глифов могут быть программно сгенерированы, при этом имя глифа не будет содержать информацию о самом символе (рис. 5).

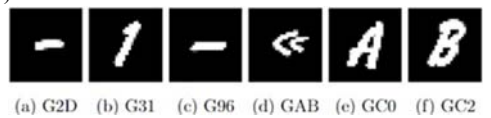

Рис. 5. Глифы из шрифта без кодировки
Fig. 5. Glyphs from a font without encoding

4.3 Распознавание изображений глифов

Распознавание изображений глифов производилось с помощью сверточной нейронной сети. Для этого сверточная нейронная сеть была обучена на наборе данных изображений глифов и использовалась для идентификации глифов в извлеченных изображениях. Код[3] для обучения нейросетевой модели был разработан с использованием платформы Google Colab [8]. Вычислительные ресурсы для обучения также предоставлялись данной платформой.

4.3.1 Подготовка данных

В рамках исследования формат изображений в наборе данных был выбран аналогично формату набора данных EMNIST [9]. Изображения (см. рис. 6) имеют следующие свойства:

- одноканальные;
- разрешение – 28x28;
- цвета инвертированы: черный фон, белый символ.

В рамках исследования был сформирован набор данных, включающий 29172 изображения, которые распределены по 102 классам. Классы представляют собой буквы русского и английского алфавитов, знаки препинания и специальные символы, например, символ ™. Для подготовки обучающей выборки были использованы изображения, сгенерированные с помощью 286 различных шрифтов. Данные были разделены на три выборки в следующем соотношении:

- Тренировочная выборка (Train) – 195 шрифтов, 20420 изображений (70%)
- Валидационная выборка (Validation) – 72 шрифтов, 7584 изображений, (26%)

[3] https://github.com/sinkudo/fonts-recognition/blob/main/ker.ipynb

- Тестовая выборка (Test) – 14 шрифтов, 1168 (4%)

Шрифты для формирования обучающей выборки были найдены в интернете в свободном доступе, после чего из каждого шрифта было извлечено 102 глифа.

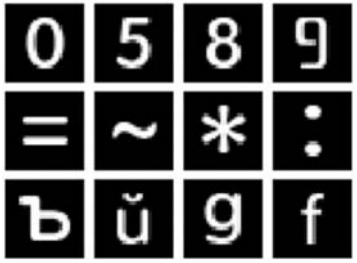

Рис. 6. Изображения из набора данных
Fig. 6. Images from the dataset

4.3.2 Аугментация данных

Для расширения обучающей выборки была проведена аугментация данных [10]. Были взяты оригинальные изображения (см. рис. 7), к которым применили такие операции (рис. 8):

- изменение масштаба (zoom);
- наложение шума (add noise);
- искажение (distortion);
- сдвиг (shear);
- поворот (rotate).

Рис. 7. Оригинальное изображение
Fig. 7. Original image

Рис. 8. Аугментированные изображения
Fig. 8. Augmented images

При обучении модели нейронной сети на аугментированной выборке точность упала на 1-2 процента из-за чего от идеи аугментации данных пришлось отказаться.

4.3.3 Архитектура сверточной нейронной сети

В листинге 1 представлены слои, используемые в модели сверточной нейронной сети.

```
Model: "sequential"
```

4.3.4 Параметры обучения

В рамках эксперимента были установлены следующие параметры:

- количество эпох – 100;
- размер порции, на которые разбиваются данные – 2000;

- скорость обучения – 0,001 (1e-3);
- оптимизатор – Adam;
- функция потерь – категориальная перекрестная энтропия (categorical crossentropy

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320
 max_pooling2d (MaxPooling2  (None, 13, 13, 32)        0
 D)
 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496
 max_pooling2d_1 (MaxPoolin  (None, 5, 5, 64)          0
 g2D)
 flatten (Flatten)           (None, 1600)              0
 dropout (Dropout)           (None, 1600)              0
 dense (Dense)               (None, 256)               409856
 dropout_1 (Dropout)         (None, 256)               0
 dense_1 (Dense)             (None, 96)                24672
=================================================================
Total params: 453344 (1.73 MB)
Trainable params: 453344 (1.73 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Листинг 1. Слои нейронной сети
Listing 1. Neural network layers

В табл. 2 представлены результаты обучения трех моделей нейронной сети, для которых использовалась метрика Accuracy (доля правильно определенных глифов). Первая модель (**Rus**) была обучена на наборе символов только русского языка, вторая (**Eng**) – только английского, а третья (**Rus+Eng**) – на наборе символов обоих языков. Модель, предназначенная для распознавания символов из обоих языков, показала результаты хуже, чем те, которые работают только с одним языком. Это может быть связано с наличием схожих по написанию символов в обоих языках – гомоглифов.

Табл. 2. Результаты обучения
Table 2. Training results

	Train	**Validation**	**Test**
Rus+Eng	78%	77%	80%
Rus	96%	89%	93%
Eng	93%	92%	94%

4.4 Сопоставление глифов с символами в тексте

После извлечения данных из PDF-документа была проведена процедура распознавания глифов и создания словаря. Ключом словаря является имя глифа, которое позволяет при необходимости получить соответствующий символ. Все изображения глифов располагаются в папке, название которой соответствует имени шрифта. Имя каждого глифа представляет собой преобразованное имя глифа в юникод символ, если это возможно. В противном случае изображение глифа сохраняется с тем же именем, что и в шрифте. Распознавание глифов осуществляется с использованием сверточной нейронной сети.

По результатам работы формируется словарь, содержащий информацию о каждом использованном шрифте (рис. 9).

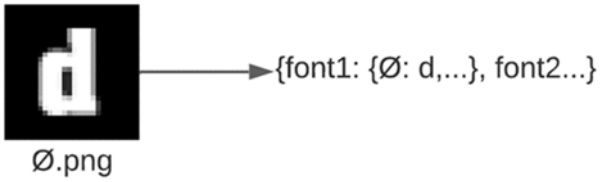

Рис. 9. Словарь соответствий глифов и символов
Fig. 9. Glyph-to-symbol correspondence dictionary

4.5 Восстановление текста

В рамках алгоритма производится последовательный анализ каждого символа текста с целью определения используемого шрифта. Шрифты можно разделить на два основных класса:

1. **Простые шрифты**, включающие TrueType, OpenType, Type1 и другие. В случае работы с этими шрифтами в тексте сохраняются коды символов, которые ссылаются на имена глифов, необходимых для отображения конкретного символа. Однако при работе с этими шрифтами может возникнуть проблема некорректной кодировки, когда вместо символа A отображается символ #.
2. **Составные шрифты**[4], такие как CID шрифты, например, Type1c, Type0, Type2. В этих шрифтах глифы отображаются с помощью CID (Character Identifier). CIDs представляют собой массив индексов от 1 до n, где n – количество глифов в шрифте, а нулевой индекс зарезервирован для символа notdef.

При копировании текста, использующего шрифт с этой проблемой, character id могут быть интерпретированы как символы Unicode. Эта проблема возникает, когда в PDF-файле отсутствует словарь, сопоставляющий CIDs с кодами символов. Для решения этой проблемы необходимо преобразовать CID в соответствующее имя глифа из шрифта. Пример восстановления текста представлен на рис. 10.

4.5.1 Дополнительный шаг по корректировке текста

В процессе работы нейронной сети могут возникать ошибки при распознавании символов, особенно в случае использования гомоглифов – похожих символов из разных алфавитов. Это может привести к необходимости корректировки текста. В табл. 3 представлены некоторые гомоглифы из русского и английского алфавитов.

Идентификация русского или английского слова в тексте возможна благодаря уникальным символам, характерным только для одного из языков.

Список уникальных символов:

- К уникальным символам русского языка относятся: я, й, ц, б, ж, з, д, л, ф, ш, щ, ч, ъ, ь, э, ю.
- Для английского языка уникальными являются следующие символы: q, w, f, u, j, l, z, s, v.

[4] https://adobe-type-tools.github.io/font-tech-notes/pdfs/5014.CIDFont_Spec.pdf

Рис. 10. Результат восстановления текста
Fig. 10. Text recovery result

Табл 3. Похожие символы
Table 3. Similar symbols

Английский	**a**	**b**	**c**	**e**	**h**	**k**	**m**	**n**	**o**	**p**	**r**	**t**	**u**	**x**	**y**
Русский	**а**	**б**	**с**	**е**	**н**	**к**	**м**	**п**	**о**	**р**	**г**	**т**	**и**	**х**	**у**

5. Оценка эффективности

5.1 Данные для тестирования

Для проведения тестирования были найдены документы, содержащие текстовый слой, который характеризуется сложным фоном [11] или структурой [12]. К документам со сложным фоном можно отнести, например брошюры, а к документам со сложной структурой газеты. Всего было собрано 14 PDF-документов. На рис. 11 представлены образцы собранных документов.

Далее из всех PDF был извлечен текст. Текст либо копировался, при условии возможности корректного копирования, либо извлекался вручную и сохранялся в файлах формата JSON. Для сравнительного анализа разработанного метода была использована библиотека Dedoc [13]. Текст из JSON файлов сравнивался с результатами, полученными с помощью библиотеки Dedoc и разработанного метода, что позволило определить эффективность последнего.

5.2 Тестирование

Для оценки качества был применен метод, основанный на метрике Левенштейна [14]. Результаты показали, что предложенный подход в среднем на 45% эффективнее по сравнению с аналогом Dedoc, который использует технологию оптического распознавания символов (OCR), что отражено в табл. 4. Для тестирования из каждого документа было взято по одной странице.

Рис. 11 Пример страниц, на которых проводилось тестирование
Fig. 11. Example pages on which testing was conducted

Табл. 4. Сравнение эффективности двух подходов.
Table 4. Comparison of the effectiveness of two approaches

	Количество символов	Dedoc	Наш подход
Integratable isolation system	1281	68%	96%
Брошюра Ростелеком	102	0%	71%
Кафетерий льгот	608	31%	99%
Брошюра StationGuard	1774	55%	99%
Спецификация военного двухканального радио	2406	62%	98%
Руководство по визуальному оформлению	130	55%	97%
Брошюра для родителей детей до 5 лет	125	0%	99%
Брошюра Biesse	347	17%	99%
Брошюра, описывающая механизм подотчетности Asia Development Bank	265	51%	99%
Exploration of Windows Vista Advanced Forensic Topics	393	72%	93%
Средство Engineer Documentation Assistant – помощник в работе со сложной документацией	2380	78%	97%
Australasian Information Security Evaluation Program	175	39%	91%
Canadian center for cyber security	166	92%	91%
Мобильные компьютеры корпоративного класса	2290	80%	97%
Средняя точность		50%	95%

При обработке документов библиотекой Dedoc были использованы параметры в табл. 5.

6. Заключение

В ходе исследования был разработан и реализован метод извлечения текста из PDF-документов, содержащих текстовый слой. Разработанный метод оптимизирован для работы с PDF-документами, имеющими нарушения в кодировке, сложный фон и плохо поддающимися стандартным методам оптического распознавания символов (OCR). Результаты исследования показали, что предложенный подход в среднем на 45% эффективнее по сравнению с аналогом Dedoc, который использует технологию оптического распознавания символов (OCR) для документов такого типа. Также была предусмотрена возможность создания пользовательских наборов данных для обучения и разработки собственной нейронной сети с поддержкой произвольного набора символов. Это позволяет адаптировать метод к различным условиям и требованиям пользователей.

Табл. 5. Параметры, использованные при извлечении текста при помощи Dedoc
Table 5. Parameters used for extracting text with the help of the Dedoc library

pdf_with_text_layer	false
document_type	other
language	rus+eng
need_pdf_table_analysis	true
is_one_column_document	auto
return_format	plain_text
need_header_footer_analysis	false

Список литературы / References

[1]. Awel M. A., Abidi A. I. Review on optical character recognition // International Research Journal of Engineering and Technology (IRJET). — 2019. — Т. 6, No 6. — С. 3666—3669.
[2]. A detailed review on text extraction using optical character recognition / C. Thorat [и др.] // ICT Analysis and Applications. – 2022. – С. 719-728.
[3]. Haralambous Y. Fonts & encodings. – "O'Reilly Media, Inc.", 2007.
[4]. Tauber J. K. Character encoding of classical languages // 2019). Digital classical philology: Ancient Greek and Latin in the digital revolution. – 2019. – С. 137-158.
[5]. Jain P., Taneja K., Taneja H. Which OCR toolset is good and why: A comparative study // Kuwait Journal of Science. – 2021. – Т. 48, No 2.
[6]. Padova T. Adobe Acrobat 8 PDF Bible. Т. 363. – John Wiley & Sons, 2007.
[7]. Smith R. An overview of the Tesseract OCR engine // Ninth international conference on document analysis and recognition (ICDAR 2007). Т. 2. – IEEE. 2007. – С. 629-633.F
[8]. Bisong E., Bisong E. Google colaboratory // Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners. – 2019. – С. 59-64.
[9]. EMNIST: Extending MNIST to handwritten letters / G. Cohen [и др.] // 2017 international joint conference on neural networks (IJCNN). – IEEE. 2017. – С. 2921-2926.
[10]. Khalifa N. E., Loey M., Mirjalili S. A comprehensive survey of recent trends in deep learning for digital images augmentation // Artificial Intelligence Review. – 2022. – Т. 55, No 3. – С. 2351-2377.
[11]. An adaptive thresholding algorithm-based optical character recognition system for information extraction in complex images / D. Akinbade [и др.] // Journal of Computer Science. – 2020. – Т. 16, No 6. – С. 784 - 801.
[12]. DocBed: A multi-stage OCR solution for documents with complex layouts / W. Zhu [и др.] // Proceedings of the AAAI Conference on Artificial Intelligence. Т. 36. – 2022. – С. 12643–12649.
[13]. Belyaeva O., Bogatenkova A., Turdakov D. Dedoc: A Universal System for Extracting Content and Logical Structure From Textual Documents //2023 Ivannikov Ispras Open Conference (ISPRAS). – IEEE, 2023. – С. 20-25.

[14]. LEVENSHTEIN V. I. // Discrete Mathematics and Applications. – 1992. – Т. 2, No 3. – С. 241–258. – DOI: doi:10.1515/dma.1992.2.3.241. – URL: https://doi.org/10.1515/dma.1992.2.3.241.

Информация об авторах / Information about authors

Михаил Викторович ЗАГОРОДНИКОВ – бакалавр направления подготовки «Прикладная информатика» Иркутского государственного университета, стажер-исследователь в молодёжной лаборатории искусственного интеллекта, обработки и анализа данных, стипендиат Института системного программирования им. В.П. Иванникова Российской академии наук. Сфера научных интересов: нейронные сети, анализ электронных документов.

Mikhail Viktorovich ZAGORODNIKOV– Bachelor's degree in Applied Informatics from Irkutsk State University, trainee researcher at the young researchers Lab of AI, Data Processing & Analysis of Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, scholarship holder of Ivannikov Institute for System Programming of the Russian Academy of Sciences. Field of scientific interests: neural networks, analysis of electronic documents.

Андрей Анатольевич МИХАЙЛОВ – заведующий молодёжной лаборатории искусственного интеллекта, обработки и анализа данных Института динамики систем и теории управления имени В.М. Матросова. Его научные интересы включают анализ электронных документов, распознавание образов.

Andrey Anatolevich MIKHAYLOV – head of the young researchers Lab of AI, Data Processing & Analysis of Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences. His research interests include document analysis, image recognition.