Сопроводительная документация по задаче:

09. Автоматизированный алгоритм обезличивания данных.

Аннотация

Разрабатываемое решение — Автоматизированный алгоритм обезличивания данных.

Целевой аудиторией решения будут сотрудники органов исполнительной власти города Москвы и подведомственных учреждений.

Конечным пользователем обезличенных документов станут компании-разработчики решений в сфере искусственного интеллекта, участники эксперимента. Также сервисом смогут пользоваться сотрудники, которым в рамках исполнения служебных обязанностей необходимо передать документы, содержащие персональные данные сторонним организациям.

1. Анализ входных данных и постановка задач

Проанализировав визуально базовый объем документов к данной задаче и имея экспертный опыт работы (создание, исполнение документов) с электронным документооборотом команда Monolith пришла к выводу: что документы в основном состоят из таблиц и текста, печатей, оттисков и прочих артефактов документооборота в организации. Текст размещен в таблицах, в шапке документа, в теле и в подвале документа, а также в таблицах без напечатанных границ. При анализе поставленной проблемы деперсонификации управленческих, распорядительных и др. документов, решение задачи виделось команде в решении двух подзадач:

1. Распознавание текста документа.
2. Лингвистический анализ получено текста с определением персональных данных.

В рамках работы над 1 подзадачей было установлено, что распознавание текста возможно при использовании библиотеки с открытым исходным кодом Tesseract, EasyOCR (https://github.com/jaidedai/easyocr), сервис визуального распознавания Yandex.Cloud, ABBYY Cloud OCR SDK(публичный API распознавания в облаке Windows Azure).

В рамках работы над 2 подзадачей было установлено, что лингвистический анализ полученного текста можно проводить с помощью таких средств как DaData, Pullenti, Abbyy Infoextractor, Dictum, Eureka, Promt, RCO, AOT, Ahunter, и открытыми решениями Natasha (https://github.com/natasha/natasha) и DeepPavlov (https://github.com/deepmipt/DeepPavlov).

2. Улучшения

По сравнению с прототипом решения, предоставленным к 1 этапу конкурса решение подверглось некоторым изменениям:

1. Осуществлен перевод распознавания ФИО с библиотеки Natasha на библиотеку DeepPavlov с моделью BERT. Это позволило добиться лучшего анализа текста по сравнению с библиотекой Natasha. Качество распознавания при помощи библиотеки Natasha изображено на рисунке 1.

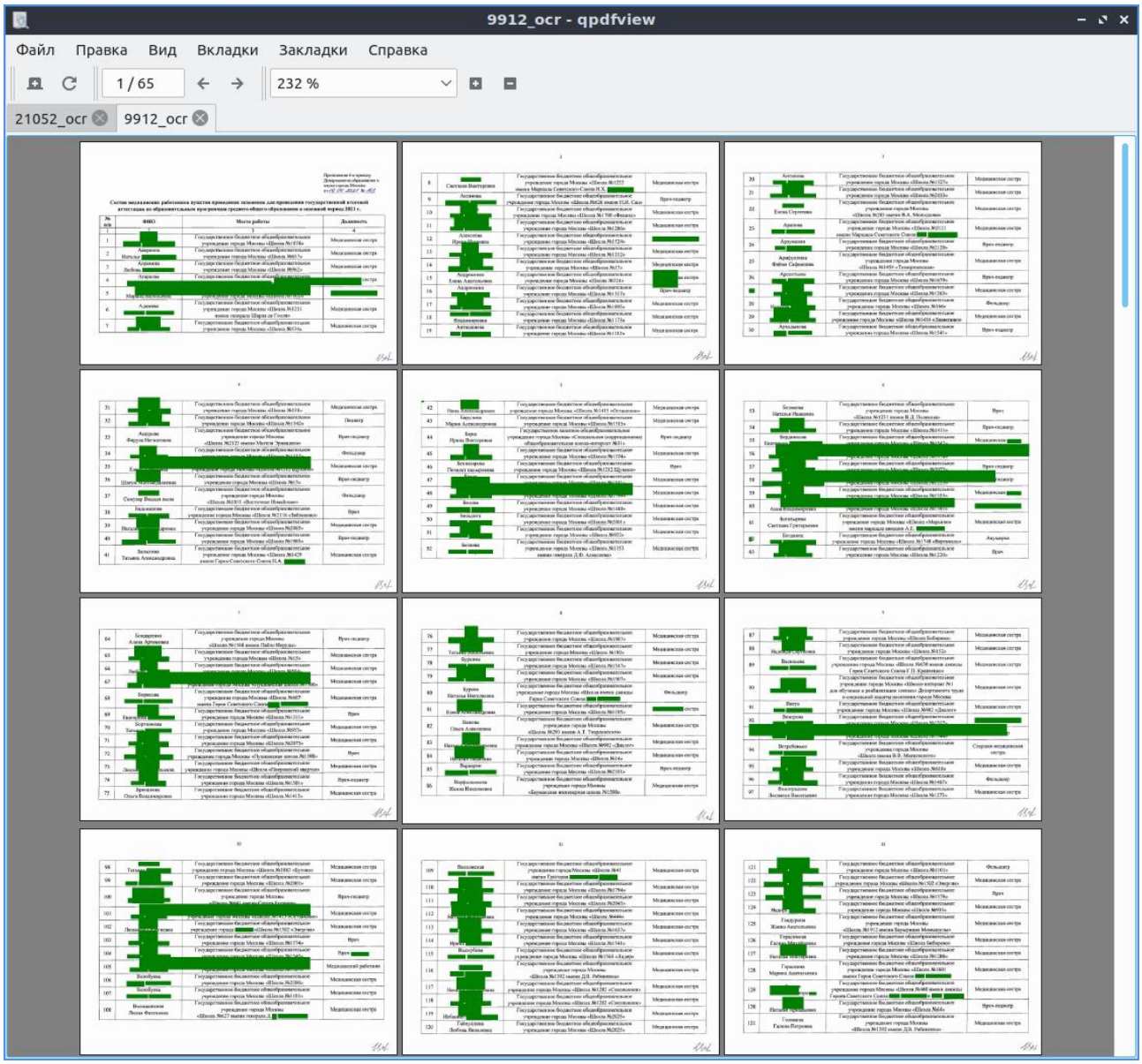

Рисунок 1. Распознавание ФИО, реализованное к концу 1 этапа конкурса.

После перевода с библиотеки Natasha на библиотеку DeepPavlov с моделью BERT качество распознавания улучшилось и приобрело вид, изображенный на рисунке 2.

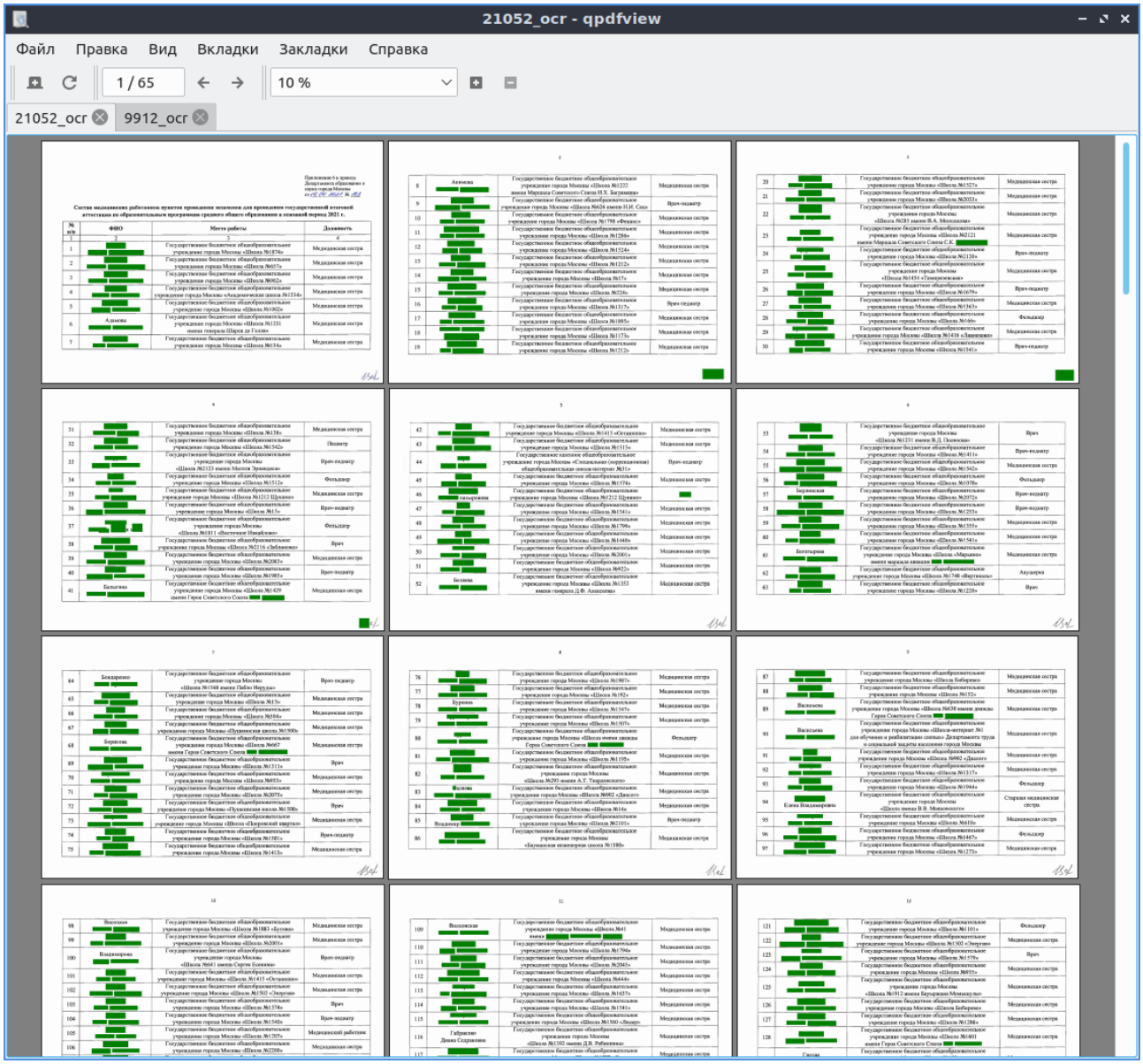

Рисунок 2. Распознавание ФИО после перевода на библиотеку DeepPavlov .

2. Осуществлен перевод функции генерации изображений JPG из PDF-документа на библиотеку pdf2image. Это позволило обеспечить лучшее качество рендера страниц, что в совокупности привело к существенному улучшению качества работы сервиса в целом.

3. Разработан анализ адресов на основе библиотеки Natasha (https://github.com/natasha/natasha) и паспортов, СНИЛС, телефонов на основе библиотеки Yargy в виде отдельного подсервиса.
4. Сервис обезличивания ФИО выведен в отдельный подсервис.

3. Выполнение задач построения решения

По состоянию на 06.11.2021 г. командой Monolith были выполнены следующие задачи построения решения:

1. Анализ требований к разработке решения согласно требований конкурса.
2. Выбор методов и средств разработки решения.
3. Анализ существующих решений в этой области.
4. Проработка способов распознавания и обезличивания ПДн в документах.
5. Разработка теоретической модели прототипа автоматизированного алгоритма обезличивания данных.
6. Проработка тестового варианта обезличивания документов на примере ФИО.
7. Проработка тестового варианта интерфейса веб-сервиса по обезличиванию документов.
8. Разработка тестовой физической реализации модели прототипа автоматизированного алгоритма обезличивания данных в виде приложения на языке программирования Python.
9. Разработка тестовой физической реализации веб-сервиса по обезличиванию документов.
10. Размещение тестовой версии приложения и его веб-интерфейса на хостинг с получением статического адреса.
11. Проверка качества работы реализованных в п. 7,8,9 тестовой версии приложения и его веб-интерфейса, включая возможность загрузки не обезличенных JPG/PDF-файлов и корректность скачивания их обезличенной версии.

12. Размещение исходного кода тестовой версии приложения на GitHub.
13. Разработка предварительной версии сопроводительной документации к решению.
14. Анализ эффективности распознавания ПДн алгоритмом и осуществление корректировки алгоритма с целью снижения число его ложных срабатываний.
15. Добавление распознавание других ПДн (таких как дата рождения, паспортные данные, телефон) в соответствии с ст. 3 Федерального закона от 27.07.2006 г. «О персональных данных» №152-ФЗ.
16. Разработка прототипа и создание веб-формы сервиса для комфортного взаимодействия его целевой аудитории, расположенной сейчас по веб-адресу: http://193.32.219.30:5000.

Таким образом, весь перечень работ, заявленный для 2 этапа конкурса, выполнен.

После проведенного анализа эффективности всех решений в области распознавания текста команда Monolith остановилась на использовании в своем решении библиотеки Tesseract как слоя распознавания, а для лингвистического анализа - использовании библиотек Natasha, Yargy и DeepPavlov (нейроная модель NER_RUS_BERT). Библиотека DeepPavlov используется для поиска ФИО в распознанном тексте, а библиотеки Natasha, Yargy используются в решении для поиска адреса, серии и номера паспорта, СНИЛС, а также телефона субъекта ПДн.

4. Лицензии

Командой Monolith было принято решение, что решение будет распространяться под лицензией Apache 2.0. Все сторонние библиотеки, используемые в решении имеют лицензии: Apache 2.0, MIT, BSD-3-Clause License и AGPL-3.0 License. Это не запрещает их использование как в личных, так и коммерческих целях.

5. Структура решения

После совещания команда пришла к выводу, что решение должно состоять из 2 подсервисов. Соответственно структура компонентов сервиса должна иметь следующий вид:

Рисунок 3. Структура решения

В дальнейшем развитии решения планируется их объединить.

6. Параметры тестового стенда

Для проверки работы решения командой Monolith был использован стенд, базирующийся в Yandex.Cloud (Compute Cloud). Он имеет следующие ресурсы:

Платформа: Intel Ice Lake

Гарантированная доля vCPU 100%

vCPU: 12

RAM: 12 ГБ

Объём дискового пространства: 20 ГБ

Прерываемая: да

GPU: нет

7. Описание сценария работы решения

Сценарий взаимодействия пользователя и сервиса имеет следующий вид:

1. Пользователь заходит в сервис обезличивания по веб-адресу: http://193.32.219.30:5000/upload и видит окно приветствия системы, где ему предлагается выбрать один из 2 подсервисов решения для выполнения своих задач. Скриншот интерфейса пользователя приводится на рисунке 4.

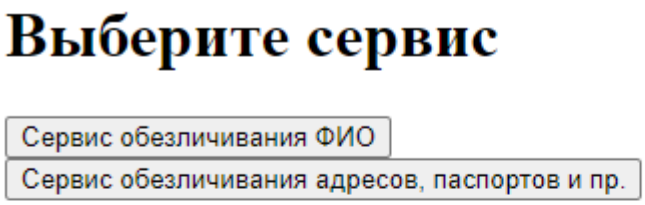

Рисунок 4. Скриншот главного экрана решения

2. Далее следует описание взаимодействия пользователя с подсервисами:

    2.1.1. Если пользователь выбирает подсервис «Сервис обезличивания ФИО», то попадает в интерфейс сервиса по обезличиванию ФИО в PDF, JPG-документах. Скриншот интерфейса пользователя приводится на рисунке 5.

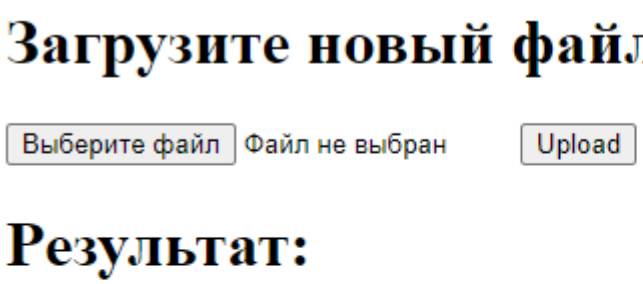

Рисунок 5. Подсервис обезличивания ФИО

2.1.2. Далее пользователь нажав на кнопку «Выбрать файл» указывает путь к файлу, который нужно обезличить.

2.1.3. После выбора файла пользователю необходимо нажать кнопку «Upload» и произойдет выгрузка указанного пользователем файла на сервер распознавания и обезличивания. Поддерживаются файлы PDF/JPG/JPEG.

2.1.4. После этого происходит конвертация PDF в массив файлов JPG. Если загружаемый на ресурс файл является JPG документом, то данный шаг пропускается.

2.1.5. После конвертации происходит передача одного изображения на распознавание текста из него.

2.1.6. После распознавания текста идет его сопоставление с координатами блоков на изображении.

2.1.7. Далее происходит затирание зеленым цветом ФИО по полученным координатам блоков.

2.1.8. Если загруженный пользователем файл на ресурс являлся PDF-документом, то происходит сборка полученных в п.2.1.4 изображений в результирующий PDF-файл. Если пользователь

загружал для обезличивания изображение(jpg/jpeg), то данный шаг пропускается.

2.1.9. Результат обезличивания отображается пользователю на ресурсе. Также пользователю доступна ссылка для скачивания обезличенного файла.

2.2. Если пользователь выбирает подсервис «Сервис обезличивания адресов, паспортов и пр.», то попадает в интерфейс сервиса по обезличиванию ФИО в PDF, JPG-документах. Скриншот интерфейса пользователя приводится на рисунке 6.

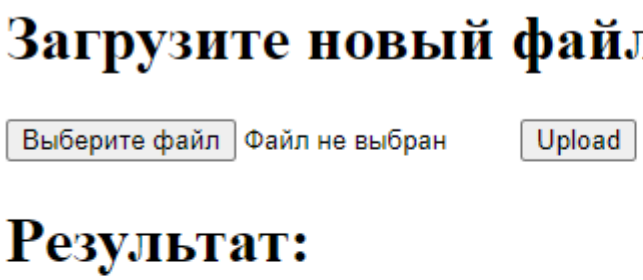

Рисунок 6. Подсервис обезличивания адресов, паспортов и пр.

2.2.1. Далее пользователь нажав на кнопку «Выбрать файл» указывает путь к файлу, который нужно обезличить.

2.2.2. После выбора файла пользователю необходимо нажать кнопку «Upload» и произойдет выгрузка указанного пользователем файла на сервер распознавания и обезличивания. Поддерживаются файлы PDF/JPG/JPEG.

2.2.3. После этого происходит конвертация PDF в массив файлов JPG. Если загружаемый на ресурс файл является JPG документом, то данный шаг пропускается.

2.2.4. После конвертации происходит передача одного изображения на распознавание текста по шаблонам из него.

2.2.5. После распознавания текста идет его сопоставление с координатами блоков на изображении.

2.2.6. Далее происходит затирание зеленым цветом адреса, СНИЛС, данных паспорта и телефонов по полученным координатам блоков.

2.2.7. Если загруженный пользователем файл на ресурс являлся PDF-документом, то происходит сборка полученных в п.2.2.3 изображений в результирующий PDF-файл. Если пользователь загружал для обезличивания изображение(jpg/jpeg), то данный шаг пропускается.

2.2.8. Результат обезличивания отображается пользователю на ресурсе. Также пользователю доступна ссылка для скачивания обезличенного файла.

8. Работа приложения

В качестве примера работы был выбран документ из 65 страниц. Его время обезличивания с момента загрузки до получения ссылки для скачивания составило 14 минут. Указанный файл был предоставлен рамках выполнения задания.

Результат работы приложения приведен на рисунке 7.

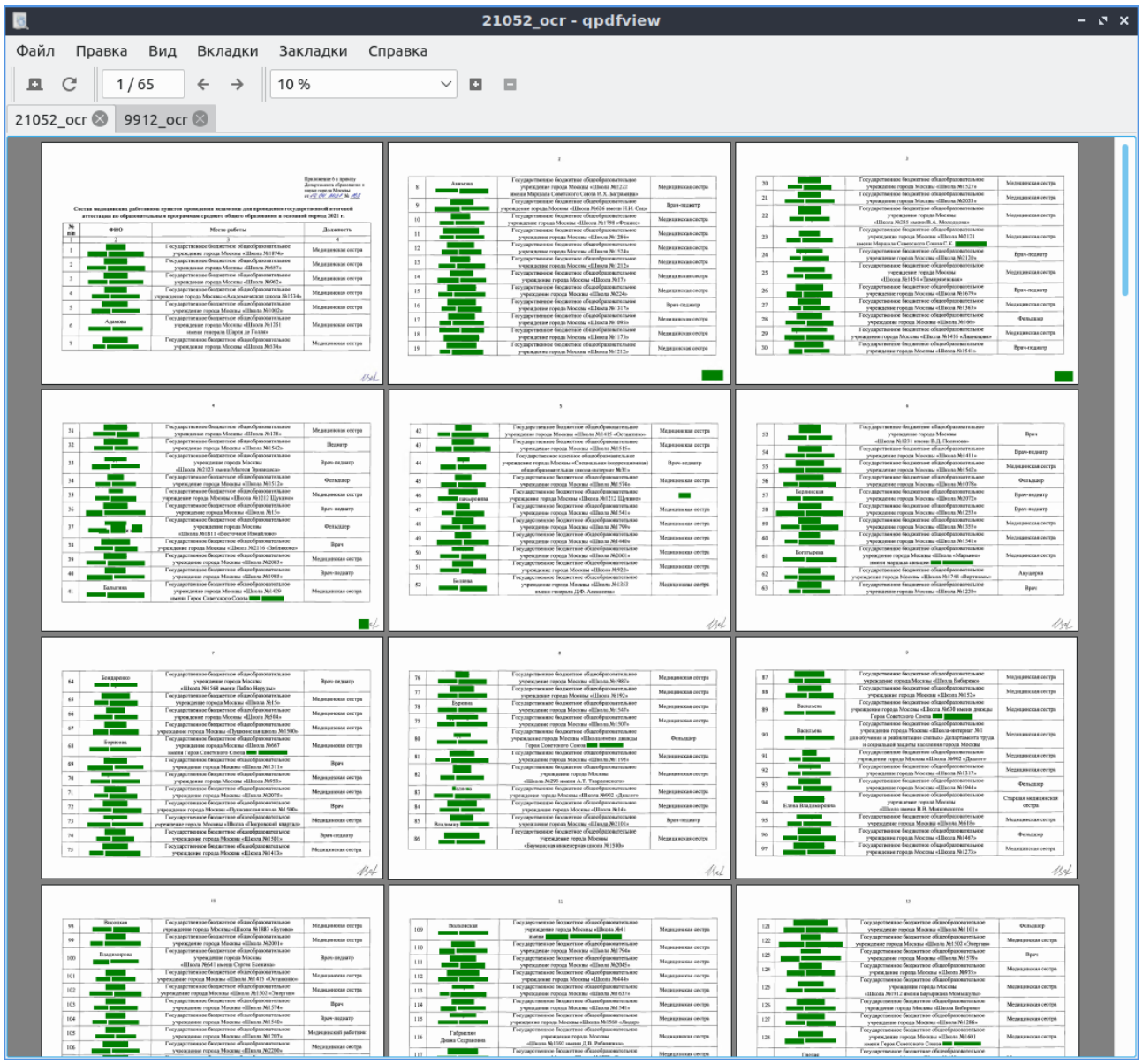

Рисунок 7. Результат работы решения.

Работа решения также была проверена на всем наборе предоставленных в рамках конкурса документов. Набор содержал более 1300 открытых служебных документов.

9. Ограничение работы решения

Для работы решения подходят PDF-документы и изображения с разрешением не менее 80 dpi.