# Создание функции распознавания текста с документов и изображений и автозаполнения форм

**Постановка задачи**

Задача: Распознавание изображений в текст (форму) + сравнение текущей формы с изображением

1. Исследование на предмет наличия инструментов (ocr, машинное обучение, другие инструменты)
2. Получить достаточно устойчивую реализацию, задача преобразовать (pdf, png, ...) в (html / xml / json)
3. Внедрение в продукт (flutter/dart)
4. Создание функции распознавания и раскидывания по форме получаемых данных

Итог: функция распознавания и автозаполнения инстанса формы

**Таблица 1. Сводная таблица технологий OCR и машинного обучения для обработки текста**

| **№** | **Технология** | **Краткое описание** | **Разработчик** | **Подход** | **Языки** | **Лицензия / Доступность** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | **Tesseract** | Один из фундаментальных open-source OCR-движков | Google (исходно HP) | Rule-based + LSTM | Многоязычный, включая кириллицу | Open-source (Apache 2.0) |
| 2 | **Google Vision API** | Облачный сервис от Google с OCR, face, logo и object detection | Google Cloud | ML + DL | Многоязычный | Коммерческий API |
| 3 | **AWS Textract** | Распознаёт текст и структуру документов, включая таблицы и формы | Amazon | ML | Многоязычный | Коммерческий API |
| 4 | **EasyOCR** | Лёгкий в использовании OCR на PyTorch | Jaided AI | DL (CNN + CRNN) | 80+ языков, включая русский | Open-source (MIT) |
| 5 | **Pytesseract** | Python обёртка над Tesseract OCR | Open-source | Rule-based (на базе Tesseract) | См. Tesseract | Open-source |
| 6 | **Keras-OCR** | OCR pipeline на Keras + TensorFlow | Fausto Morales | DL (ResNet, CRNN) | Многоязычный (обучаемый) | Open-source |
| 7 | **OpenCV + Tesseract** | Комбинированный подход: предобработка с OpenCV + Tesseract для распознавания | Комьюнити | Комбинированный | См. Tesseract | Open-source |
| 8 | **Azure Computer Vision** | Облачный сервис с OCR и визуальной аналитикой | Microsoft Azure | ML + DL | Поддерживает кириллицу | Коммерческий API |
| 9 | **OCRopus (ocropy)** | OCR библиотека, построенная поверх Tesseract, с возможностью обучения | Thomas Breuel | ML (LSTM) | Многоязычный | Open-source |
| 10 | **PaddleOCR** | Современный DL-фреймворк от Baidu, поддерживает сложную верстку | Baidu | DL (CNN, RNN, Transformer) | Многоязычный, вкл. кириллицу | Open-source |
| 11 | **DocTR** | Современный OCR pipeline от Mindee на базе PyTorch | Mindee | DL (Transformers + CNN) | Встроенные модели — англ., можно дообучить | Open-source |
| 12 | **Yandex Vision** | Сервис компьютерного зрения от Яндекса | Яндекс | ML + DL | Поддерживает кириллицу, русский | Коммерческий API |

**Таблица 2. Сводная таблица лицензирования OCR-технологий**

| **Лицензия / Тип** | **Коммерческое использование** | **Модификация** | **Распространение** | **Требуется указание авторства** | **Требуется открытие модификаций** | **Примечания** |
| --- | --- | --- | --- | --- | --- | --- |
| **Apache 2.0** (Tesseract) | Разрешено | Да | Да | Да | Нет | Удобна для встраивания в закрытые решения |
| **MIT** (EasyOCR, Pytesseract, DocTR, PaddleOCR и др.) | Разрешено | Да | Да | Да | Нет | Очень простая и гибкая лицензия |
| **BSD-like** (OCRopus) | Разрешено | Да | Да | Да | Нет | Похож на MIT, подходит для продакшена |
| **Коммерческие API** (Google Vision, AWS Textract, Azure, Yandex Vision) | По условиям соглашения | Нет доступа к коду | Нет | Не применяется | Нет | Использование возможно только через API, плата за запросы, нельзя модифицировать |

**Обзор пайплайнов для распознавания текста с документов**

В статье Сбера описывается подход к созданию единой модели для распознавания 7 типов документов (паспорт, ИНН, СНИЛС и др.) 1 Аналогичный подход можно применять для медицинских документов:

1. Модель классификации: Определяет тип документа (анализ крови, мочи и т.д.)
2. Определение угла поворота: Корректирует ориентацию документа
3. Сегментация: Находит расположение всех полей на документе
4. Распознавание текста: Извлекает текст из выделенных областей
5. Постпроцессинг: Исправление ошибок, проверка по справочникам

* **Проблемы и решения**

Основные проблемы при распознавании медицинских документов:

1. Разнообразие форматов: Медицинские документы могут сильно отличаться между учреждениями. Решение - создание гибких шаблонов или использование подходов с обучением на разнообразных данных.
2. Качество изображений: Плохое освещение, размытие, артефакты. Решение - предварительная обработка изображений (бинаризация, увеличение контраста) с помощью OpenCV или аналогичных библиотек
3. Специфическая терминология: Медицинские термины, сокращения, коды МКБ. Решение - использование специализированных словарей и дообучение моделей на медицинских текстах.
4. Рукописный текст: Записи врачей часто трудночитаемы. Решение - применение ICR (Intelligent Character Recognition) технологий или гибридных подходов.

* **Интеграция с клиническими исследованиями**

OCR-технологии могут быть интегрированы в процесс клинических исследований для:

1. Автоматического извлечения данных из медицинских карт и результатов анализов
2. Обработки информированных согласий (проверка полноты и корректности заполнения)
3. Анализа протоколов исследований и извлечения ключевых параметров
4. Сравнения результатов обследования в динамике

* **Выводы**

Для сложных случаев (рукописные заметки врачей) потребуется комбинация нескольких технологий, возможно, с привлечением ICR.

Важно предусмотреть этап валидации результатов распознавания, особенно для критически важных данных (дозировки лекарств, результаты анализов).

Для увеличения точности можно использовать гибридные подходы, например, сначала классифицировать тип документа, затем применять специализированную модель для его обработки, как это делалось в Сбере

Дальнейшее исследование направлено на сравнение точности различных OCR-решений конкретно на медицинских документах и разработку специализированных моделей для разных типов медицинских форм.

**Текущий пайплайн для тестирования распознавания текста с документов, используемых в клиническом исследовании и разработки baseline-модели**

Baseline-модель (базовая модель) — это простая исходная модель, которая используется как точка отсчёта для сравнения с более сложными или оптимизированными алгоритмами в машинном обучении и data science.

Цели применения:

1. Оценка эффективности – помогает понять, есть ли смысл усложнять модель.
2. Быстрая проверка гипотез – позволяет убедиться, что задача вообще решаема выбранными методами.
3. Минимальный рабочий вариант (MVP) – если baseline уже дает приемлемый результат, его можно использовать в продакшене.

Baseline в текущем исследовании:

1. Применение easyocr, ocrmypdf, pdf2image, zero-shot-classification на bert
2. Сравнение результатов для 5 документов разного качества и 5 изображений разного качества.
3. Минимальный рабочий вариант (MVP) – интеграция решения для оценки совместимости с текущими версиями библиотек и проектом, производительности (например, при использовании onnx-формата) и времени работы модели, подсчет метрик и оценка интерпретируемости результата.

**EasyOCR 2**

EasyOCR — Python-библиотека для OCR, построенная на базе PyTorch.

Плюсы:

* Простота использования (минимальный код для начала работы)
* Поддержка множества языков «из коробки»
* Хорошая точность на стандартных документах
* Не требует предварительной обработки изображений

Минусы:

* Менее точна на сложных документах по сравнению с коммерческими решениями
* Требует значительных вычислительных ресурсов для больших объемов данных

В рамках текущей задачи может быть полезной для быстрого прототипирования решений по распознаванию медицинских документов.

**OCRmyPDF 3**

OCRmyPDF — это инструмент на основе Python, который добавляет слой OCR к PDF-файлам, сохраняя исходное форматирование. Использует Tesseract OCR в качестве бэкенда, но предоставляет удобную обертку с дополнительными функциями для работы с PDF.

Плюсы:

* Сохранение структуры PDF – не пересоздает документ, а добавляет OCR-слой поверх исходного содержимого
* Поддержка многостраничных PDF – автоматически обрабатывает все страницы
* Оптимизация выходных файлов – сжатие изображений, удаление дубликатов
* Гибкость настроек – можно регулировать качество OCR, языки, предобработку изображений
* Поддержка CLI и Python API – удобен как для скриптов, так и для ручного использования

Минусы:

* Зависит от Tesseract – если Tesseract плохо справляется с документом, качество OCRmyPDF тоже будет низким
* Не всегда хорошо работает с рукописным текстом
* Медлкенный при обработке больших объемов

Применение в текущей задаче:

Может использоваться для распознавания отсканированных медицинских PDF, чтобы сделать их текстово-поисковыми без изменения исходной разметки.

**pdf2image4**

pdf2image– **у**тилита для конвертации PDF в набор изображений (JPEG/PNG). Полезна для предварительной подготовки документов перед передачей в OCR-движки, которые не работают напрямую с pdf.

Плюсы:

* Простота использования – минимум кода для извлечения страниц PDF в виде картинок
* Поддержка высокого DPI – можно задавать разрешение для улучшения качества OCR
* Гибкость вывода – сохраняет каждую страницу как отдельное изображение или объединяет в один файл

Минусы:

* Не выполняет OCR самостоятельно – только подготовка изображений для других инструментов
* Может занимать много места при конвертации больших PDF в изображения

Применение в текущей задаче:

Конвертирует PDF в изображения, которые далее используются в EasyOCR.

**Zero-shot classification на BERT 5**

Zero-shot-классификация — это подход, позволяющий классифицировать тексты по категориям, которые модель не видела во время обучения. В основе лежат предобученные языковые модели типа BERT, которые:

* Используют семантическое понимание текста за счет предобучения на больших корпусах
* Формулируют задачу классификации как задачу textual entailment (текстового следствия)
* Оценивают совместимость входного текста с каждой из предложенных категорий

Таким образом, входной текст медицинского документа сопоставляется с каждой преустановленной категорией («анализ крови», «информированное согласие» и т.д.), модель вычисляет вероятность, что текст соответствует описанию категории.

В контексте клинических исследований и обработки медицинских документов zero-shot подход может быть использован для первичной классификации документов или фильтрации по релевантности (автоматическое отсеивание неклинических документов, выделение документов, относящихся к конкретному исследованию)

Этот метод может быть также первым этапом финального пайплайна – триггеринг разных OCR-конвейеров для разных типов документов.

**Почему Zero-shot?**

Проблема отсутствия датасетов для клинических исследований. Публичные датасеты медицинских документов отсутствуют в связи с коммерческой и этической чувствительностью данных, невозможностью использовать реальные данные для обучения без деидентификации.

Последствиями могут трудности с адаптацией предобученных моделей. Возможно использование синтетических данных, применение трансферного обучения с общемедицинских датасетов, итеративный процесс улучшения.

**Ограничения метода**

* Точность ниже, чем у специализированных обученных моделей
* Зависимость от формулировок категорий — нужно тщательно подбирать названия классов
* Проблемы с короткими текстами
* Вычислительная стоимость для больших объемов документов

**Особенности работы zero-shot classification с медицинскими документами**

1. Специфическая терминология: медицинские аббревиатуры (МРТ, КТ, ОАК) и термины, коды МКБ и другие стандартизированные обозначения. BERT-модели, дообученные на медицинских текстах (BioBERT, ClinicalBERT), показывают лучшие результаты.
2. Структурные особенности: можно "запомнить" шаблонные фразы (референсные значения, заключение), табличные данные в результатах анализов. BERT может лучше работать на текстовой части документов, чем на числовых данных, это тоже нужно учитывать
3. Контекстная зависимость: одно и то же слово может относиться к разным категориям в зависимости от контекста. Например, "гемоглобин" может встречаться и в направлении на анализ, и в результате, в шапке документа могут быть указания на другие документы.

**Обзор выполненной работы по реализации baseline-модели**

Был разработан и протестирован пайплайн для обработки медицинских документов с использованием различных технологий OCR. Работа включала следующие этапы:

1. Настройка рабочего окружения и установка необходимых зависимостей
2. Реализация трех основных подходов к распознаванию документов:

* Прямая обработка PDF с помощью OCRmyPDF
* Распознавание текста с изображений через EasyOCR
* Конвертация PDF в изображения с последующим распознаванием

1. Эксперименты с классификацией типов документов

1. Установка и настройка

Были установлены и настроены следующие ключевые компоненты:

* Tesseract OCR
* easyocr
* Ghostscript и Poppler-utils (для работы с PDF)
* Python-библиотеки:
* pytesseract
* pdf2image
* ocrmypdf

Рабочее пространство организовано в четыре директории:

* input\_papers - для исходных PDF-документов
* output\_papers - для обработанных PDF с OCR-слоем
* input\_images - для изображений с текстом
* from\_pdf\_images - для изображений, конвертированных из PDF

2. Реализованные методы обработки

2.1. Прямая обработка PDF (OCRmyPDF)

Реализована обработка PDF-документов с помощью OCRmyPDF, которая добавляет текстовый слой поверх исходного документа, сохраняя его структуру. Пример использования:

Рис.3 Фрагмент кода для конвертации pdf

ocrmypdf --force-ocr input.pdf output.pdf

Наблюдения:

* Хорошо работает с чистыми сканами
* Обработка занимает несколько минут на файл (30-страничный документ обрабатывался долго)
* Не всегда корректно работает с документами сложной структуры

2.2. Распознавание текста с изображений (EasyOCR)

Реализована функция ocr\_easyocr(), которая использует EasyOCR для распознавания текста с изображений.

Рис.4 Фрагмент кода распознавания изображений easyocr

reader = easyocr.Reader(['ru'])

def ocr\_easyocr(image):

    result = reader.readtext(image, detail=0)

    return " ".join(result)

images\_list = []

dict\_text = {}

input\_folder\_path = "/content/input\_images"

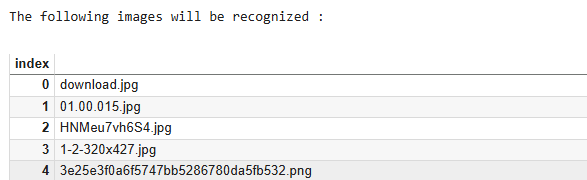
for filename in os.listdir(input\_folder\_path):

    if filename.endswith(("pdf", "png", "jpg")):

      images\_list.append(filename)

print("The following images will be recognized : ")

pd.DataFrame(images\_list, columns=["Filename"])

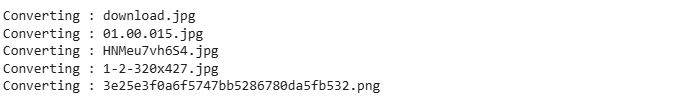


for image in images\_list:

  print("Converting : " + image)

  input\_filename = input\_folder\_path+'/'+image

  dict\_text[image] = ocr\_easyocr(input\_filename)



dict\_text['3e25e3f0a6f5747bb5286780da5fb532.png']

Министерство Здравоохранения Код формы по ОКУД и Социального развития РФ Код учреждония по ОКПО Наименованио учреждения Медицинская документация Форма & 224/у Лаборатория Утв. Минздравом СССР 04.10.80 Ыз 1030 АНАЛИЗ КРОВИ ^2 20 дата взятия биоматериапа Фамилия; И.О: Возраст Учреждение Участок медицинская карта ^ Норма Резуль Единицы; подлежащие -тат Единицы СИ замене Гемоглобин М Ж 130,0-160,0 ГЕЛ 13,0-16,0 г%о 120,0-140,0 12,0-14,0 Эритроциты МЖ 4,0-5,0 \*10'2/л 4,0-5,0 МЛН: в 1 мм 3,947 (мкл) Цветовой показатель 0,85-1,05 0,85-1,05 Среднее содержание гемоглобина (в 30-35 30-35 ПГ эритроците) Ретикулоциты 2-10 %о 2-10 %о Тромбоциты 180,0-320,0 \*1О9/л 180,0- ТЫс- в мм 320,0 (мкп) Лейкоциты 4,0-9,0 \*10"/л 4,0-9,0 ТЫс в мм (мкл) Миопоциты % % \*1О"рл в 1 мм' (мкл) МетамИ лоциты % % \*1О",л ммз (мкп) 1 Палочкоядерные 1-6,30040- \*10л 40-60 мм" (мкп) Сегментоядерные 47-72 2,000- % 47-72 % 5,500 \*1О9/л 2000-5000 в 1 мм' (мкп) Эозинофипы 0,5-5 0,020- % 0,5-5 % 0,300 \*1ОЧл 20-300 мм? (мкп) Базофилы % 0-1 % 0-1 0-0,065 \*109/л 0-65 (мкп) Лимфоциты 19-37 1,200- % 19-37 % 3,000 \*1О"/л 1200-3000 мм' (мкл) Моноциты 3-11 0,090- 3-11 % 0,600 \*10\*/л 90-600 мм3 (мкл) Плазматические клетки % % \*109/л мм' (мкл) Скорость (реакция) М 2-10 2-10 ммеч ммчас Оседания эритроцитов Ж 2-15 2-15 ммз %

* Отличные результаты на качественных сканах (точность ~90%)
* Плохие результаты на фотографиях документов
* Среднее время обработки: около 2 минут на 5 файлов
* Поддерживает русский язык

3.3. Конвертация PDF в изображения с последующим распознаванием

2.3 Реализован альтернативный подход:

* Конвертация PDF в набор изображений (pdf2image)
* Последовательное распознавание каждого изображения (EasyOCR)

Результаты:

* Позволяет обрабатывать сложные PDF-документы
* Дает возможность предварительной обработки изображений
* Требует больше временных и вычислительных ресурсов

3. Классификация документов

Проведены эксперименты по автоматической классификации документов с использованием zero-shot подхода на основе модели facebook/bart-large-mnli.

Рис.5 Фрагмент кода zero-shot классификации

dict\_pdf\_text['Индивидуальная регистрационная карта пациента\_page\_1.jpg']

Индивидуальная регистрационная карта пациента протокола научного исследования (Оценка психического здоровья населения в условиях коронавирусной эпидемиих Инициалы ФИО: рождения: Пол: Дата время подписания пациентом добровольного информированного согласия на участие в научном исследовании: Раздел 1 Ваше образование: среднее: 2 профессиональное неоконченное высшее 3 высшее 4 учёная степень 5 Семейное положе- официальный брак ние: гражданский брак 2 не женат не замужем 3 Трудовая занятость: ПО найму государственных учреждениях ПО найму в частных учреждениях . 2 свой бизнес 3 безработный/ая получаю образование 4 не работаю Место Вашего фак- Санкт-Петербург & тического прожива- Ленинградская область . НИЯ: иное 3 Диагностировал ЛИ биполярное аффективное расстройство 2 врач у Вас когда- депрессивное расстройство либо психические генерализованное тревожное/паническое/фобическое рас- расстройства: стройство: 3 заболевания шизофренического спектра и другие психические расстройства. 4 нет 5 Имеются ли у Вас со- эндокринологические (сахарный диабет, ожирение; гипо/ги- матические заболева- пертиреоз и др ). 1 нИЯ: кардиологические (гипертония; стенокардия, аритмия и др. 2 респираторные (бронхит, астма; ХОБЛ): 3 желудочно-кишечные (гастрит, панкреатит, колит, ЖКБ и др. ): 4 мочеполовые (пиелонефрит, цистит, гломерулонефрит и др. 5 неврологические (мигрень, энцефалопатия; инсульт и др: 6 нет Было Л и у Вас под- да; мазок Из носа/горла (ПЦР) ' тверждено заболева- да, КТ признаки поражения ние СОУ--19: только клиника ОРВИ 3 здоров/ва Получали ЛИ Вы ле- да; требовалась госпитализация 4 чение от СОУ-19: да; лечился на дому . 2 нет не лечился 3 Был ли диагностиро- нет ван СОУ--19 у ва- да, У родственников 2 ших близких: да; у друзей 3 да; у коллег 4 Год

text\_from\_easyocr = dict\_pdf\_text['Индивидуальная регистрационная карта пациента\_page\_1.jpg']

document\_types = ["анализ", "исследование", "протокол", "индивидуальная регистрационная карта"]

text = text\_from\_easyocr

result = classifier(text, document\_types, multi\_label=False)

predicted\_type = result["labels"][0]

print(f"Тип документа: {predicted\_type}")



Результаты:

* Текущая реализация показывает нестабильные результаты
* Модель часто ошибается, даже когда в тексте есть явные указания на тип документа
* Требуется доработка (few-shot learning или другие подходы)

4. Выявленные проблемы и пути решения

1. Низкое качество распознавания фотографий документов. Необходимо добавить этап предобработки изображений (коррекция освещения, поворот, увеличение контраста)
2. Долгая обработка многостраничных PDF. Решение: Реализовать параллельную обработку страниц.
3. Плохая классификация документов. Решение: использовать специализированные медицинские модели (BioBERT), создать небольшой размеченный датасет для fine-tuning, добавить rule-based классификацию по ключевым словам
4. Потеря порядка страниц при конвертации PDF в изображения, реализовать систему нумерации и сортировки файлов

5. Планы по дальнейшему развитию

* Реализовать модуль автоматической и ручной предобработки изображений
* Добавить поддержку большего количества типов медицинских документов
* Разработать систему валидации результатов
* Оптимизировать производительность: кэширование загруженных моделей, пакетная обработка документов

**Заключение**

Разработанный пайплайн позволяет обрабатывать медицинские документы различных типов, но требует доработки. Наилучшие результаты достигнуты при обработке качественных сканированных документов. Для повышения точности и надежности системы необходимо реализовать дополнительные модули предобработки и постобработки, а также улучшить систему классификации документов. По результатам реализации baseline-модели принято решение о дальнейшей разработке собственного решения для программного обеспечения «Inmedikum.Online»