

docreader-ocr

Python-библиотека для автоматического
распознавания российских документов

Паспорт РФ · СНИЛС · Аттестат · Диплом

Авторы: Кардаш Михаил Михайлович, Абзелилов Руслан Эдуардович

Руководитель: к.п.н., доцент кафедры БИИИТ, Карманова Екатерина Владимировна

ФГБОУ ВО «МГТУ им. Г.И. Носова» · 2026

```
pip install docreader-ocr
```

Зачем это нужно – и почему сейчас

3–10 мин

время ручного ввода
одного пакета документов

~150 ч

рабочих часов в год
на одного оператора

до 5%

ошибок при ручном
вводе персональных данных

Проблема: Любая организация, работающая с людьми, ежедневно тратит сотни часов на ручной перенос данных из документов в информационные системы. Это медленно, дорого и порождает ошибки.

Цель проекта: Разработать Python-библиотеку, которая автоматически извлекает структурированные данные из фотографий российских документов и встраивается в любую существующую систему.

С 2022 года государство ускорило цифровизацию: ФГИС, электронные очереди, госуслуги. Большинство организаций оцифровали фасад – заявление подаётся онлайн, но внутри сотрудник всё равно вручную перебивает данные из документа. docreader-ocr закрывает этот разрыв.

Задачи: классифицировать тип документа · детектировать поля · распознать текст · вернуть структурированный результат · обеспечить соответствие 152-ФЗ

Структура репозитория docreader-ocr

```
src/docreader/  
  classifier/  
    base.py  
    yolo_classifier.py  
  detector/  
    base.py  
    yolo_obb.py  
  ocr/  
    base.py  
    easyocr_engine.py  
  resolver/  
    base.py  
    lvl_resolver.py  
  preprocessing/  
    geometry.py  
  pipeline.py  
  factory.py  
  hub.py  
  config.py  
  schemas.py  
tests/  
  eval/  
  run_eval.py  
  preannotate.py  
pyproject.toml  
README.md
```

classifier/

YOLO11n-obb: находит все документы на фото, вырезает OBB-кроп каждого

detector/

Отдельная YOLO OBB-модель на каждый тип документа. Детектирует 8–10 полей

ocr/

EasyOCR (VGG-BiLSTM-CTC) с кастомным дообучением под русские документы

resolver/

Различает аттестат/диплом по OCR поля lvl + fuzzy matching (rapidfuzz)

pipeline.py

Главный DocReader: оркестрирует все компоненты в единый конвейер

hub.py

Авто-загрузка весов из GitHub Releases в ~/.cache/docreader/models/

Трёхуровневый ИИ-конвейер

1

Классификатор
YOLO11n-obb

Тип документа
ОВВ-круп
Асс = 97.5%



1a

Resolver
EasyOCR + fuzzy

Только для
attestat/diplom
по полю lvl



2

Детектор зон
YOLO OBB x4

8–10 полей
per doc type
mAP@50=90%



3

OCR
EasyOCR ft

VGG-BiLSTM-CTC
Word Acc=87.3%
~30 мс/поле



dict
Результат

{'surname':
'Абзелилов', ...}

```
from docreader import DocReader

result = DocReader().process("passport.jpg")

print(result.documents[0].fields) # → {"surname": "Кардаш", "passport_num": "1234 567890", ...}
```

Почему библиотека, а не API: данные не покидают инфраструктуру · соответствие 152-ФЗ · интеграция 2 строками кода · нет абонплаты · IT-отдел контролирует всё

Формирование датасетов

Классификатор	Детектор зон	OCR
Синтетика (Faker) + реальные фото с согласия субъектов	До 200 фото на тип документа, разметка в Label Studio (OBB)	11 752 примера (строка + изображение поля)
Классы: passport, snils, diplom/attestat, prilozhenie	4 независимых датасета (passport, snils, attestat, diplom)	Синтетика: рендер на фонах из реальных кропов
Split: 70/20/10%	Формат: YOLO OBB	Faker: ФИО, даты, адреса, номера документов
Аугментации: rotate $\pm 30^\circ$, perspective, blur, JPEG	Синтетика: Faker + шаблоны с аугментациями	Алфавит: кириллица, латиница, цифры, спецсимволы

Четыре итерации – от идеи до production

v1	Классификатор MobileNetV2 5 классов	Детектор YOLO11n-det (bbox)	OCR EasyOCR base	Примечание Нет поддержки нескольких документов. Нет OBB.
v2	Классификатор MobileNetV2 5 классов	Детектор YOLO11n-obb	OCR EasyOCR base	Примечание Классификатор всё ещё не видит несколько объектов.
v3	Классификатор YOLO11n-obb 5 классов	Детектор YOLO11n-obb	OCR EasyOCR fine-tuned	Примечание Attestat и diplom визуально идентичны – модель путает.
v4 ✓	Классификатор YOLO11n-obb 4 класса	Детектор YOLO11n-obb	OCR EasyOCR fine-tuned	Примечание Production. Resolver разделяет по полю lvl.

Архитектура нейросетей и параметры обучения

Классификатор и детектор – YOLO11n-obb

Функция потерь: Rotated IoU Loss + BCE + Distribution Focal Loss
Оптимизатор: auto (AdamW) · Epochs: 100 · Batch: 64 · imgsz: 640
Предобучение: COCO · Устройство: CUDA
Модель сходится к 40–50 эпохе – задача визуально простая

OCR – VGG → BiLSTM → CTC

Transformation: None · FeatureExtraction: VGG (256 каналов)
SequenceModeling: BiLSTM (hidden=256) · Prediction: CTC
Оптимизатор: Adadelata (lr=1.0, rho=0.95) · 30 000 итераций
Fine-tuning от cyrillic_g2.pth · Время обучения: ~52 мин

Почему YOLO OBB, а не det?

Документы бывают повёрнуты.
OBB точнее обтягивает поле –
кроп для OCR становится чище.
IoU вырос с ~0.78 до 0.96

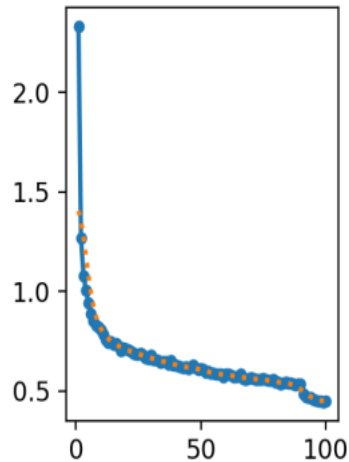
Почему EasyOCR, а не PaddleOCR?

До дообучения EasyOCR лучше.
После: EasyOCR 87.3% vs
PaddleOCR ~79%.
Проще конфигурировать (YAML)

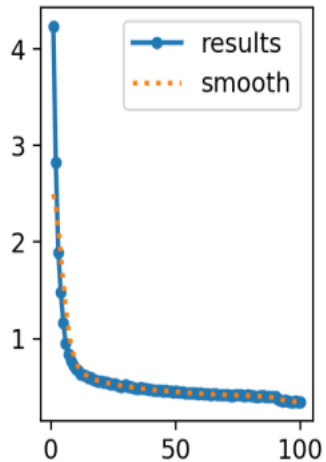
Почему CTC Loss?

Не требует посимвольного
выравнивания – достаточно
указать строку целиком.
Работает с переменной длиной

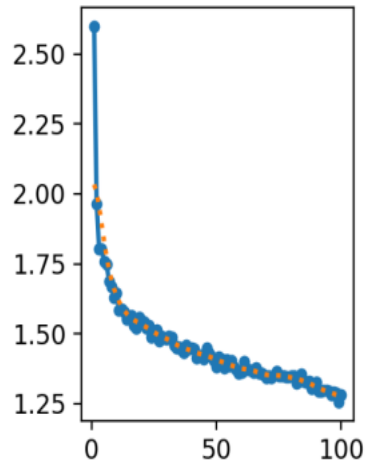
train/box_loss



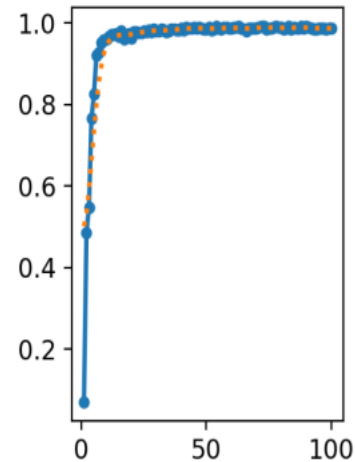
train/cls_loss



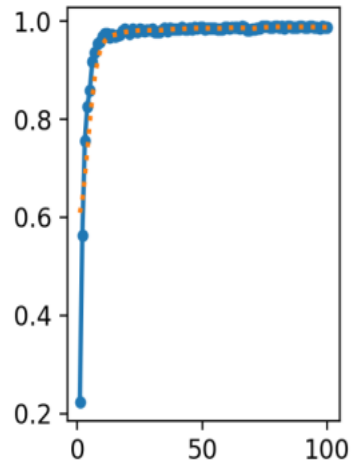
train/dfl_loss



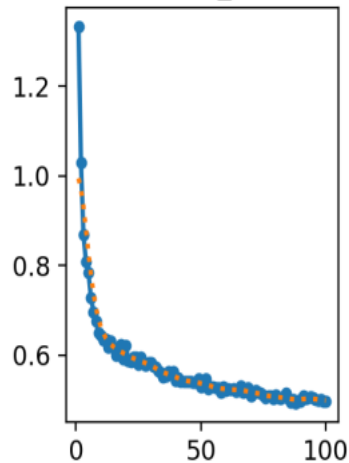
metrics/precision(B)



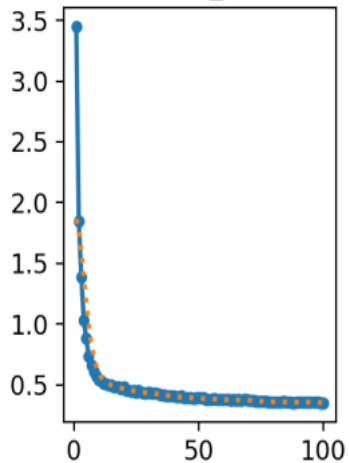
metrics/recall(B)



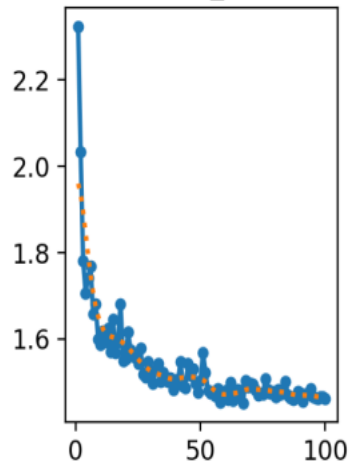
val/box_loss



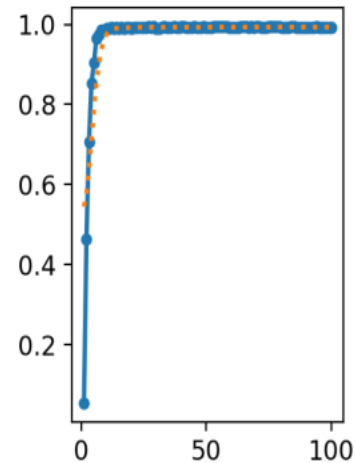
val/cls_loss



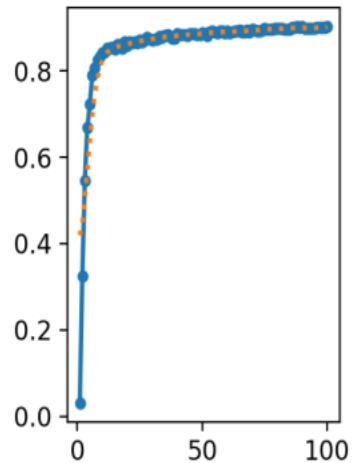
val/dfl_loss



metrics/mAP50(B)



metrics/mAP50-95(B)



EasyOCR Training Metrics (Final Report)

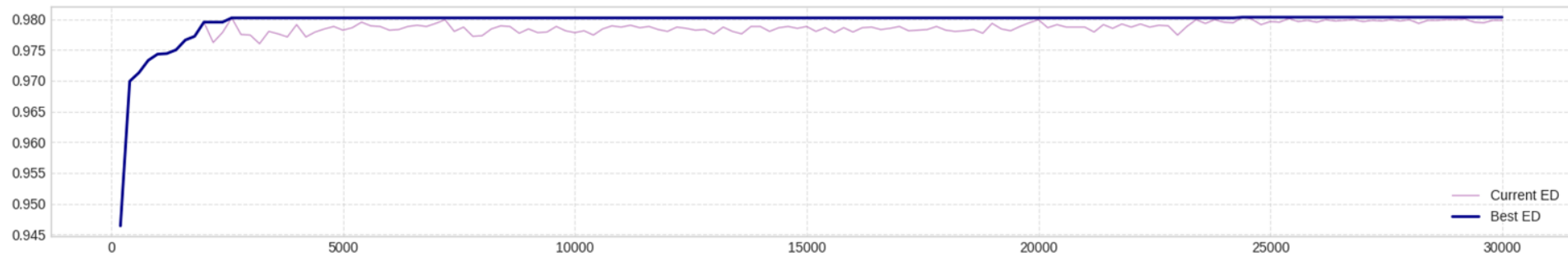
Loss Functions



Accuracy (%)



Normalized Edit Distance



Метрики качества на тестовой выборке

97.5%

Accuracy
классификатора

90.0%

mAP@50
детектора зон

87.3%

Word Accuracy
OCR (fine-tuned)

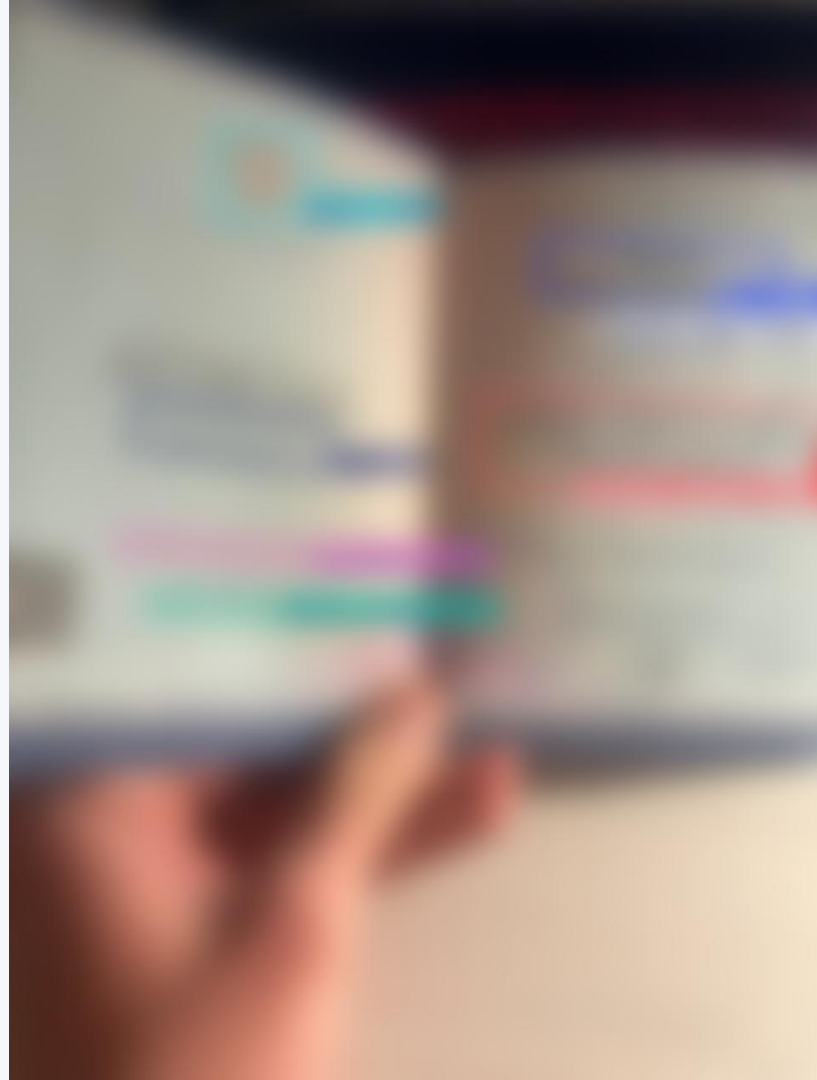
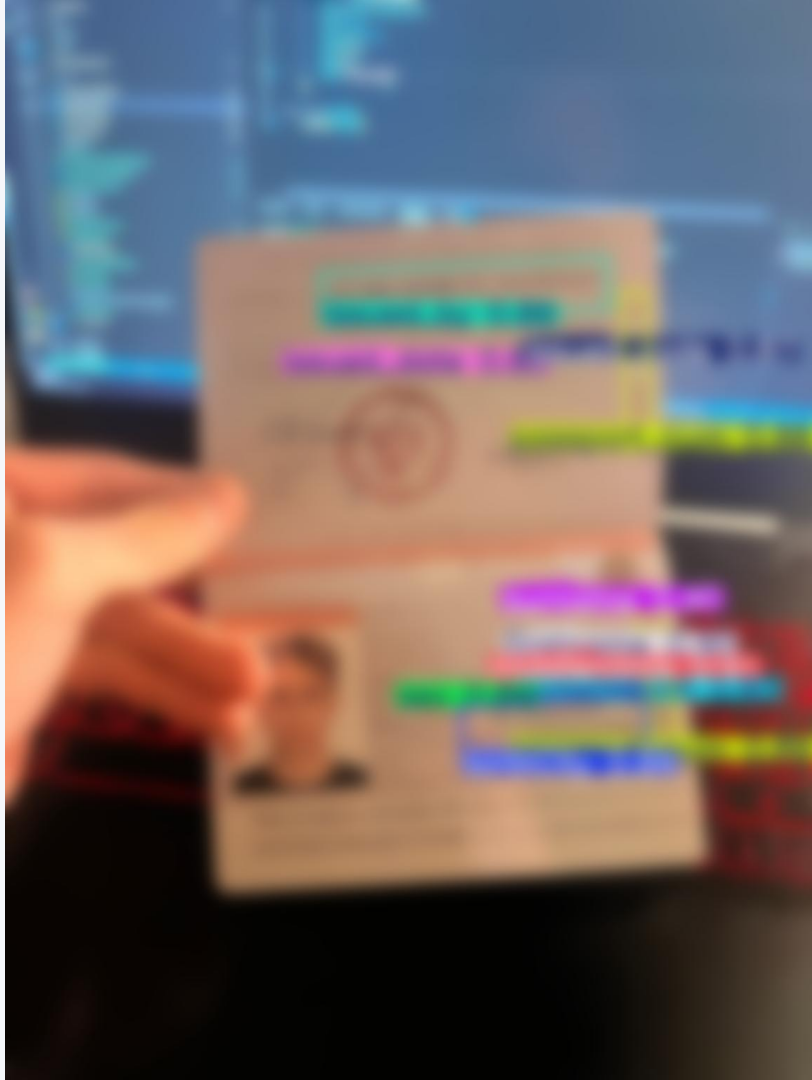
15–30с

Время обработки
одного документа

Зона	Precision	Recall	F1	IoU	N
birth_city	100.0%	96.7%	98.3%	92.3%	30
dateOfBirth	81.3%	89.7%	85.3%	89.5%	29
fio	92.2%	96.5%	94.3%	98.9%	86
lvl	98.3%	98.3%	98.3%	99.4%	57
Issued_by	96.8%	100.0%	98.4%	95.0%	30

Тип документа	Full Doc Acc.	Field Acc.	Документов	Полей
attestat	50.0%	78.0%	30	91
diplom	0.0%	48.6%	27	105
passport	6.7%	28.2%	30	163
snils	62.1%	64.4%	29	45

Бизнес-метрики: Field Accuracy (доля верных полей): 49–78% в зависимости от типа · Full Document Accuracy: 30% · Система – СППР: оператор получает большинство полей предзаполненными и верифицирует, а не вводит с нуля



Три строки кода – и система работает

```
from docreader import DocReader

result = DocReader().process("passport.jpg")

print(result.documents[0].fields)
```

152-ФЗ

Данные не покидают инфраструктуру.
Никаких внешних серверов.

Любая система

1С, CRM, ERP, мобильное приложение – без переписывания процессов.

Расширяемость

Новый тип документа = дообучить модель
+ строка в конфиге. Без вендора.

Без абонентской платы

В отличие от облачных API – никаких лимитов и ежемесячных платежей.

Новый сценарий: онбординг клиентов

При регистрации банк/страховая автоматически распознаёт документ → данные сразу попадают в CRM → система предлагает персонализированный продукт на основе профиля клиента. Время онбординга сокращается с 15 до 2–3 минут.

Новый сценарий: верификация подлинности

Библиотека фиксирует геометрию полей и типографику. Отклонение от эталонного шаблона – сигнал для ручной проверки. Основа для системы выявления поддельных документов.

Три строки кода – и система работает

А сейчас мы покажем вам видео и демонстрацию работы

Но пока мы их запускаем, вот вам милого котика



Люди, которые это сделали



Кардаш Михаил

Специалист по большим данным ООО «ММК-Информсервис»

Выпускник ФГБОУ ВО «МГТУ им. Г.И. Носова» профиля

«Искусственный интеллект в цифровой экономике»

Обучение моделей детекции полей

Обучение модели распознавания символов

Разметка фотографий

Формирование библиотеки



Абзелилов Руслан

Инженер-программист ООО «ОСК»

Выпускник ФГБОУ ВО «МГТУ им. Г.И. Носова» профиля

«Искусственный интеллект в цифровой экономике»

Обучение модели классификации

Разметка фотографий

Формирование библиотеки

Проведение экспериментов с моделями

Что дальше

Высокий

Постобработка поля «пол»

Словарь допустимых значений (МУЖ/ЖЕН). Прогнозируемый Exact Match: с 43% до ~90%.

Средний

Расширение датасета OCR

С 11 752 до 50 000+ примеров – снизит переобучение, поднимет Exact Match.

Средний

Новые типы документов

ИНН, водительское удостоверение, медсправки, трудовая книжка.

Новый

Детекция фейковых документов

Анализ геометрии и типографики полей.
Отклонение от эталона – сигнал проверки.

Новый

Онбординг и персонализация

Распознал документ → данные в CRM → персональное предложение клиенту.

docreader-ocr

Распознавай. Не вводи вручную.

```
pip install docreader-ocr
```

PyPI: <https://pypi.org/project/docreader-ocr/>

GitHub: <https://github.com/mishanyacorleone/docreader>