## **Техническое задание**

### **Цель проекта**

Создание Python-сервиса для автоматического извлечения информации из изображений документов (паспорт РФ, ПТС) и формирования ответа в формате JSON.

### **Требования к сервису:**

#### **Функциональные требования:**

1. Сервис должен принимать фотографии документов:  
   * Паспорт гражданина РФ
   * Паспорт транспортного средства (ПТС)
2. Сервис должен распознавать следующую информацию:

**Из паспорта гражданина РФ:**

* Фамилия, имя, отчество
* Серия и номер паспорта
* Дата выдачи паспорта
* Орган, выдавший паспорт
* Дата рождения
* Кем выдан (название органа)

**Из паспорта транспортного средства:**

* Серия и номер документа
* Наименование транспортного средства
* Год выпуска транспортного средства
* Дата выдачи документа
* Орган, выдавший документ
* VIN-код

1. Результат работы сервиса – JSON ответ, содержащий все распознанные данные в четко структурированном виде.

Пример структуры JSON-ответа:

{

"passport": {

"series": "4510",

"number": "123456",

"surname": "Иванов",

"name": "Иван",

"patronymic": "Иванович",

"birth\_date": "01.01.1990",

"passport\_issued\_date": "01.01.2010",

"passport\_issued\_by": "ОВД района Пресненский г. Москвы"

},

"vehicle\_passport": {

"document\_number": "77УО 123456",

"vehicle\_name": "Toyota Camry",

"vehicle\_year": "2020",

"document\_issue\_date": "10.01.2022",

"document\_issued\_by": "ГИБДД УМВД по г. Москве",

"vin\_code": "1HGCM82633A004352"

}

4. Сервис должен иметь обработку ошибок:

- Плохое качество изображения

- Необходимые поля не распознаны

- Некорректный формат документов

5. Технические требования:

- Язык программирования: Python

- Использование OCR (например, Tesseract или EasyOCR)

- Использование библиотек OpenCV, pytesseract или аналогичных инструментов

- API, построенный с использованием FastAPI или Flask для удобной интеграции с другими сервисами.

5. Сервис должен включать логику предварительной обработки изображений (обрезка, поворот, улучшение качества), чтобы повысить точность распознавания.

6. Документация API должна содержать описание endpoint’ов и формата запросов/ответов.

Критерии успешности проекта:

- Точность распознавания текста не менее 95% при хорошем качестве фото

- Среднее время обработки одного изображения не должно превышать 10 секунд

- Готовый сервис должен быть легко масштабируемым и интегрируемым в текущие системы.

Итоговые материалы разработки:

- Исходный код на Python

- Dockerfile для контейнеризации приложения

- Подробная документация по развертыванию и использованию сервиса