

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

## **На разработку OCR-системы для документов низкого качества сканирования**

### **1. Основание для разработки**

Разработка системы осуществляется в рамках выполнения выпускной квалификационной работы студента 4-го курса ИВТ 2.2 Магера Егора Владимировича на тему «Разработка OCR-системы для документов низкого качества сканирования».

### **2. Цель и назначение разработки**

**Цель:** Повышение точности и скорости автоматического распознавания текста в отсканированных документах с низким качеством изображения.

**Назначение:** Система предназначена для автоматического распознавания текста в документах, полученных при помощи сканирования с дефектами (низкое разрешение, засветы, размытость, шумы, низкая контрастность), с последующим преобразованием в редактируемые форматы.

### **3. Технические требования к системе**

#### **3.1. Требования к функциональным характеристикам**

Система должна обеспечивать:

- Загрузку изображений документов в форматах: JPEG, PNG, TIFF, PDF.
- Предварительную обработку изображений: повышение резкости, удаление шумов, коррекция контраста и яркости, коррекция перспективы.
- Распознавание текста на русском и английском языках с возможностью расширения на другие языки.
- Экспорт распознанного текста в форматы: TXT, DOCX, PDF (с текстовым слоем).

- Визуализацию результатов распознавания с выделением областей текста и возможностью ручной корректировки.
- Логирование процессов обработки и выгрузки отчетов о точности распознавания.

### **3.2. Требования к надежности**

- Система должна обеспечивать сохранность исходных и обработанных документов.
- При возникновении ошибок (например, поврежденный файл) система должна информировать пользователя без прекращения работы.
- Реализация механизма автоматического сохранения промежуточных результатов обработки.

### **3.3. Условия эксплуатации**

- Серверная часть должна работать под управлением ОС Linux или Windows.
- Для реализации системы должны использоваться: Python 3.9+, библиотеки OpenCV, Tesseract OCR, PyTorch/TensorFlow (по выбору разработчика).
- Клиентская часть — веб-интерфейс на основе Flask/Django, доступный через современные браузеры.

### **3.4. Требования к аппаратному обеспечению**

- Для работы системы требуется устройство с доступом в Интернет и современным браузером.
- На серверной стороне рекомендуется не менее 4 ГБ ОЗУ и 2 ядер CPU для обработки документов в реальном времени.

### **3.5. Требования к информационной и программной совместимости**

- Система должна поддерживать интеграцию с облачными хранилищами (Google Drive, Yandex Disk) по API.
- Должна быть обеспечена совместимость с СУБД PostgreSQL для хранения метаданных документов.

- Возможность развертывания в Docker-контейнере.

### **3.6. Требования к программной документации**

В состав документации должны входить:

- Исходный код системы.
- Пояснительная записка.
- Руководство пользователя и администратора.
- Техническое описание архитектуры и используемых алгоритмов.

### **4. Техничко-экономические показатели**

Внедрение системы позволит:

- Сократить время на ручной ввод текста из сканированных документов.
- Увеличить точность распознавания по сравнению с существующими OCR-решениями для низкокачественных изображений.
- Снизить трудозатраты на обработку архивных и поврежденных документов.

### **5. Стадии и этапы разработки**

1. Анализ требований и исследование аналогов — 2 недели.
2. Проектирование архитектуры и выбор технологий — 2 недели.
3. Разработка ядра OCR и модуля предобработки изображений — 6 недель.
4. Разработка веб-интерфейса и интеграция — 3 недели.
5. Тестирование, отладка, написание документации — 3 недели.
6. Сдача проекта — 1 неделя.

### **6. Порядок контроля и приемки**

Контроль осуществляется по завершении каждого этапа. Приемка системы осуществляется на основании соответствия функциональности требованиям настоящего ТЗ, наличия полной документации и успешного тестирования на реальных документах низкого качества.