

# Описание программы

## 1. ОБЩИЕ СВЕДЕНИЯ

Программа разработана в соответствии с планом программы переподготовки «Искусственный интеллект в промышленности».

Разработанная программа распознавания подписей – «Система распознавания подписи» (SRS – Signature Recognition System) – предназначена для идентификации владельца рукописной подписи по сканированному образу.

Программа написана на языке Python с использованием фреймворка: FEDOT (платформа с открытым исходным кодом для решения задач автоматического моделирования и машинного обучения) и библиотеки OpenCV (открытый код с алгоритмами общего назначения, компьютерного зрения и обработки изображений).

Репозиторий с кодом программы можно изучить по адресу: <https://kurl.ru/vCii>. Для его использования необходим компилятор Visual Studio Code или PyCharm, папка со сканированными образами подписей и таблицей базы данных, содержащей ФИО возможных владельцев подписей с индексами в соответствии их фамилии по алфавиту.

## 2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

### 2.1 Назначение программы

Задача программы - распознать рукописную подпись на сканированном изображении и идентифицировать ее владельца. Для успешного решения поставленной задачи выполняются следующие операции:

- создание датасета;
- определение области печати с подписью;
- выделение признаков подписи человека на документе, скрытой информационным шумом (печатью);
- подготовка изображений:
  - обрезка изображения по контуру,
  - изменение размеров изображения,
  - переворачивание изображения,
- удаление цифрового шума:
  - удаление текста,
  - удаление печати,
- классификация подписи.

- идентификация подписи по имеющейся базе данных с другими подписями людей.

## **2.2 Классы решаемых задач**

- преобразование входных данных под необходимый формат: запись всех идентификаторов владельцев подписей латиницей, конвертация всех идентификаторов в один csv-файл, конвертация всех подписей в один csv-файл, преобразование PDF-файла в изображение;
- подготовка датасета: создание, автоматический сбор, генерация, сохранение пути;
- хранение подготовленных изображений подписи в виде MatLike;
- цветовая фильтрация изображения, позволяющая отделить подпись с печатью;
- обнаружение и удаление круглой печати алгоритмом Хафа;
- восстановление утраченных фрагментов подписи;
- обрезка изображения по контуру подписи;
- обучение нейронной сети по заранее подготовленным изображениям;
- проверка точности нейронной сети на основе тестовой выборки;
- сохранение и загрузка нейронной сети в/из локальных файлов компьютера;
- интерпретация выходных данных нейронной сети для дальнейшей классификации изображения.

## **2.3 Функциональные ограничения на применение**

Для корректной работы SRS следует учитывать важные ограничения на применение:

- на изображении должна быть чётко видна только одна подпись, которую пользователь хочет идентифицировать;
- изображение должно быть в формате png, jpg, bmp или pdf;
- круглая печать на изображении должна быть явно видна;
- изображение не должно быть черно-белым.

## **2.4 Область применения**

Данное решение позволяет вести работу с различными документами и деловыми бумагами. Оно применимо в сфере реализации государственных услуг, юриспруденции, в учреждениях медицинского и образовательного направления, в бизнесе и других сферах, в которых необходимо обрабатывать большие объёмы документации (для проверки принадлежности подписи на документе заявленному лицу).

Разработанная программа – это универсальная система для проверки принадлежности подписи на документе заявленному лицу, которую можно будет приспособливать под конкретные нужды заказчика. Потенциальными клиентами, которые непосредственно будут работать с

программой, могут являться сотрудники из различных сфер профессиональной деятельности: частный бизнес – секретарь, делопроизводитель, финансист, специалист отдела кадров, офис-менеджер; государственное учреждение – бухгалтер, администратор; нотариальная контора – нотариусы и т.д.

### **3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ**

#### **3.1 Используемые методы**

Используемые алгоритмы решения поставленных задач являются в большей мере базовыми алгоритмами библиотек OpenCV и FEDOT.

Преобразование изображения, включая подготовку, устранение цифрового шума, а также конвертацию, полностью обеспечивают базовые функции OpenCV.

Основной моделью нейронной сети является глубокая сверточная нейронная сеть FEDOTCNN, позволяющая производить более глубокий анализ изображения. Реализация нейронной сети обеспечена базовым функционалом библиотеки FEDOT.

#### **3.2 Алгоритм(ы) программы**

На рисунке 1.2.1 представлена блок-схема алгоритма работы основного модуля.



Рисунок 1.2.1 – Алгоритм работы основного модуля

На рисунке 1.2.2 представлена блок-схема алгоритма работы модуля извлечения подписи с изображения.

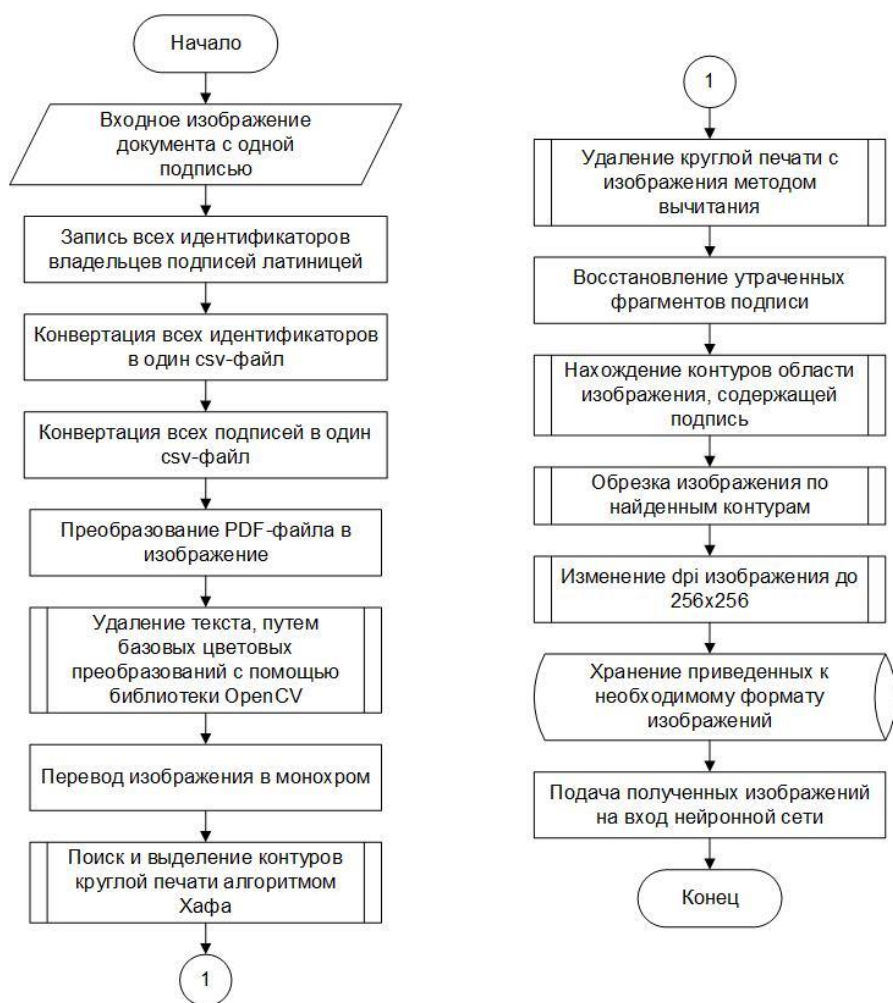


Рисунок 1.2.2 – Алгоритм работы модуля извлечения подписи с изображения

На рисунке 1.2.3 представлена блок-схема алгоритма работы модуля идентификации владельца подписи.



Рисунок 1.2.3 – Алгоритм работы модуля идентификации владельца подписи

### 3.3 Структура программы

Структура программы представляет собой диаграмму классов разрабатываемой программы (см. рисунок 1.3.1). Диаграмма состоит из 5 классов: PictureClass, StampClass, DateClass, AutographClass, ClassificationClass.

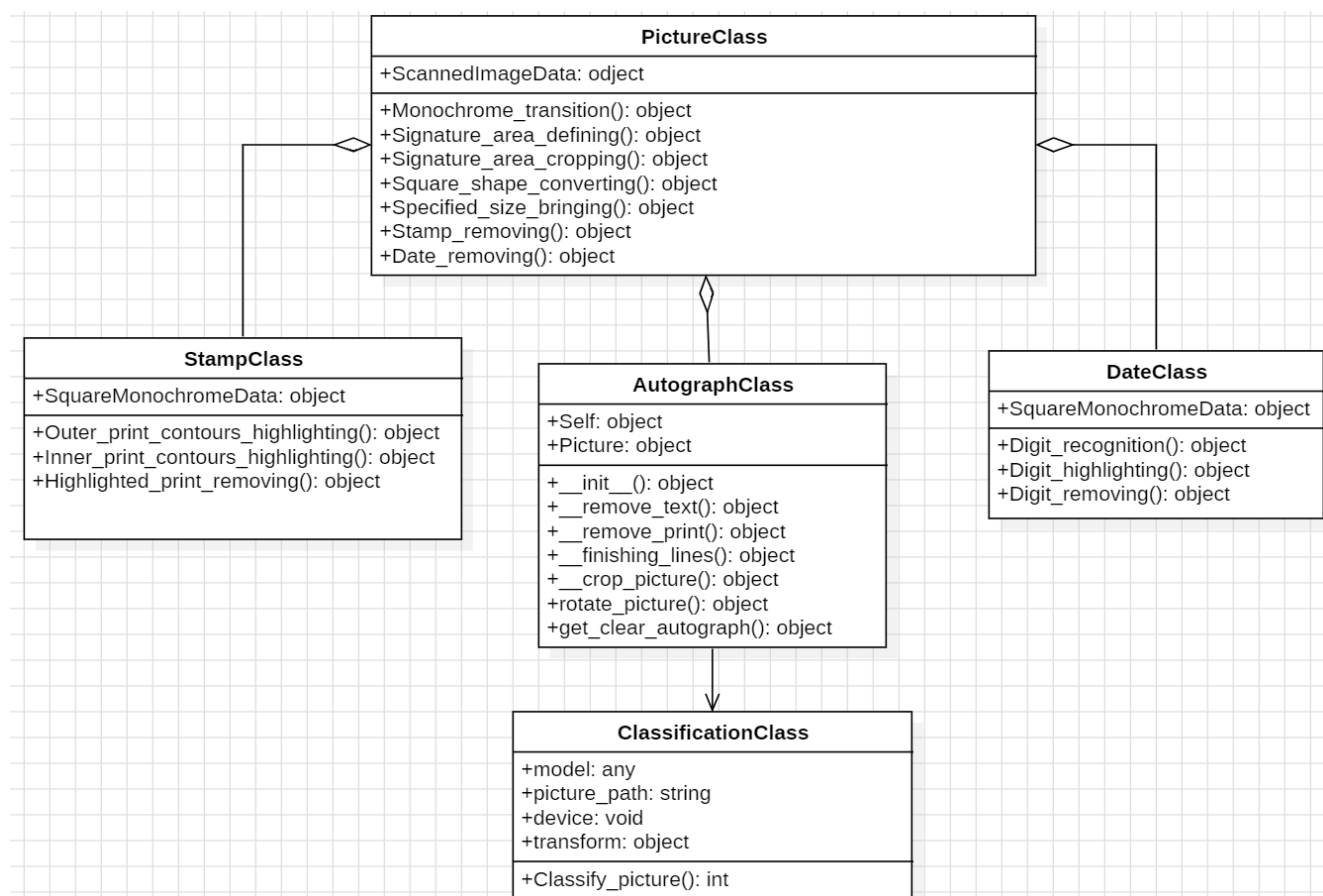


Рисунок 1.3.1 – Диаграмма классов SRS

**Класс «PictureClass»** имеет один атрибут – скан-образ документа с подписью и печатью (ScannedImageData) – а также несколько методов для работы с ним: переход в монохромный режим (Monochrome\_transition), определение области с подписью (Signature\_area\_defining), обрезание области с подписью по контуру (Signature\_area\_cropping), преобразование обрезанного скан-образа в квадратную форму (Square\_shape\_converting), приведение к нужному разрешению (Specified\_size\_bringing), удаление печати с области с подписью (Stamp\_removing) и удаление даты с области с подписью (Date\_removing). Данный класс обрабатывает и подготавливает входное изображение для дальнейшей работы с ним и идентификации подписи. Данный класс состоит из трёх подклассов – классов «StampClass», «DateClass» и «AutographClass».

**Класс «StampClass»** хранит в себе атрибут – квадратный монохром (SquareMonochromeData), преобразованный от изначального входного изображения путём перевода в чёрно-белый градиент, обрезания области с печатью и подписью и перевода в квадратный

формат размером NxN пикселей. Этот класс также осуществляет методы к атрибуту: выделение внешних контуров (`outer_print_contours_highlighting`), выделение внутренних контуров печати (`inner_print_contours_highlighting`), удаление выделенной печати (`highlighted_print_removing`). Все методы нужны для выделения и удаления с подписи информационного шума в виде круглой печати.

**Классу «DateClass»** передаётся один атрибут – квадратный монохром (`SquareMonochromeData`), почти тот же, что и в классе «StampClass», но без печати. Используемые классом методы: распознавание цифр (`Digit_recognition`), выделение цифр (`Digit_highlighting`), удаление цифр (`Digit_removing`). Все методы применяются для выделения и удаления информационного шума в виде прописанной даты.

**Класс «AutographClass»** содержит два атрибута – исходный загруженный файл с подписью и печатью (`Self`) и объект `MatLike` из `OpenCV` (`picture`). Методы для работы с имеющимися данными состоят из: подготовки входного изображения – приведение к нужному размеру, выделение круглой печати – (`__init__`), удаления текста с изображения (`__remove_text`), удаления круглой печати (`__remove_text`), восстановления утраченных фрагментов подписи (`__finishing_lines`), обрезания изображения по контуру подписи (`__crop_picture`), задания правильной ориентации скан-образу (`__crop_picture`), получения обработанной подписи (`get_clear_autograph`).

**Класс «ClassificationClass»** имеет несколько атрибутов: обученная модель из `FEDOT`, путь к классифицируемому изображению (`picture_path`). `Classify_picture()` – метод, позволяющий получить идентификатор объекта, которому принадлежит подпись на сканированном образе. Идентификация происходит уже после подготовки, выделения подписи и её распознавания.

На рисунке 1.3.2 представлена диаграмма компонентов системы.

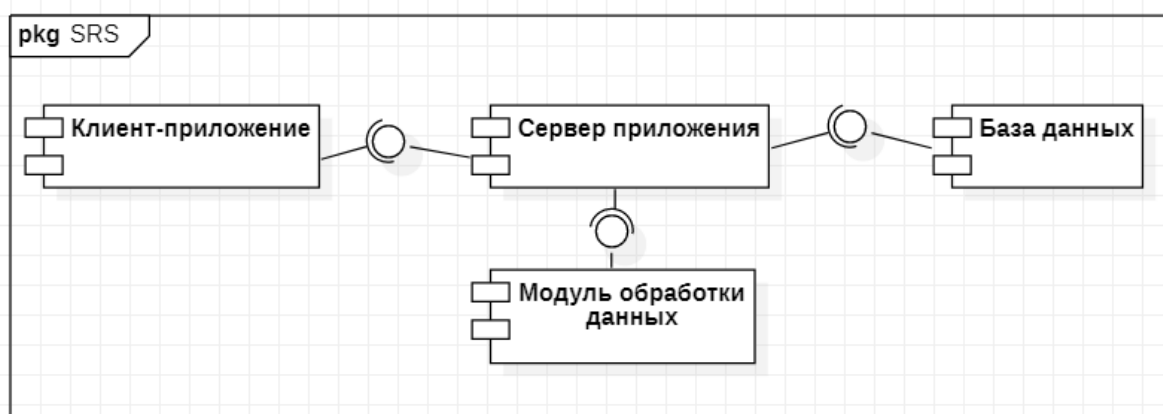


Рисунок 1.3.2 – Диаграмма компонентов

Система состоит из четырех основных компонентов:

- Клиент-приложение – блок, содержащий в себе интерфейс пользователя.

- Сервер приложения – блок, управляет доступом к данным и транзакциями.
- База данных – блок хранения преобразованных изображений.

Модуль обработки данных – основной блок системы, содержащий в себе набор функций и нейронных сетей, предназначенных для выполнения основной функции программы: распознавание скрытой подписи.

#### **4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА**

Вычислительная система под управлением ОС Windows/Linux со следующими характеристиками:

- процессор: Intel\AMD-совместимый, не менее 2\4 ядер, не менее 2.5\3 ГГц, соответственно;
- видеокарта: NVIDIA, не старше 7 поколения (GeForce GTX 750 Ti), для AMD не старше Rx 300 (R9 380x);
- оперативная память: не менее 8 Гб;
- дисковая подсистема: не менее 250 Гб;
- пропускная способность сетевых интерфейсов: не менее 1 Гбит/с.

Для сканирования анализируемого файла обязательно иметь сканер, подключенный к устройству посредством кабеля или беспроводной сети.

#### **5. ВЫЗОВ И ЗАГРУЗКА**

##### **5.1 Базовые функции программы**

На рисунке 5.1.2 представлен основной модуль системы, отвечающий за сборку датасета и вызов методы классификации изображения. Для этого подключаем библиотеки `signess` и `inskrub`, более полные названия приведены ниже на рисунке 5.1.1. Делаем инициализацию переменных, хранящих в себе путь к папке с подписями и печатью одного человека, к датасету, к файлу `csv` с остальными подписями и к входному изображению в формате `png`. Сначала собирается датасет с помощью класса `Dataset`, в который передаются нужные параметры. После этого создается не-обученная нейросеть при помощи метода `FedotCNN`. Остаётся только собрать пайплайн обучения модели и вызвать метод классификации изображения.

```
from signess.dataset import Dataset
from signess.network import FedotCNN

from inskrib.autograph import Autograph
from inskrib.documents import Document
```

Рисунок 5.1.1 – Подключаемые библиотеки

```
if __name__ == '__main__':
    set_random_seed(1)

    # config
    path_to_data = './example/docs'
    path_to_picture = './result/autographs/0-first_person-0.png'
    path_to_save_and_load = './model'

    path_to_dataset = create_dataset(path_to_data)
    create_network(path_to_dataset, path_to_picture, path_to_save_and_load)
    load_network(path_to_dataset, path_to_picture, path_to_save_and_load)
```

Рисунок 5.1.2 – Конфигурация и вызов главной функции программы в основном модуле

## 5.2 Пайплайны на основе функций компонента.

На рисунке 5.2.1 представлен пайплайн системы.



```

# create Network
network = FedotCNN()
# load dataset where path - path to .npz file
dataset = network.load_dataset(path=path_to_dataset)
# train model with loaded dataset by 3 epochs
network.train(dataset=dataset, num_epochs=3)

# predicts dataset
predicts = network.predict(dataset)
print(f'Predicts: {predicts}')

# classify picture
predict = network.classify(path_to_picture, path_to_dataset)

```

Рисунок 5.2.1 – Пайплайн обучения модели

## 6. ВХОДНЫЕ ДАННЫЕ

### 6.1 Состав и структура входных данных

Наименование данных	Обозначение	Структура данных	Способ ввода данных	Ограничения
Исходный файл с подписью и печатью	self	Документ в классе AutographClass	Загрузка документа с подписью	Документ не чёрно-белый; тип файла pdf, jpg, png или bmp
Датасет с подписями	dataset	Таблица csv с изображениями подписей в классе Data	Загрузка изображений с подписью	csv файл
Объект MatLike из OpenCV	picture	Изображение MatLike в классе AutographClass	Загрузка изображений с подписью	

### 6.2 Подготовка входных данных

Входные данные должны соответствовать следующим условиям:

- документ с подписью отсканирован;
- отсканированное изображение имеет тип png, jpg, bmp или pdf
- на отсканированном изображении находится только одна единственная подпись.

Исходный файл с подписью загружается из локальной папки.

## 7. ВЫХОДНЫЕ ДАННЫЕ

### 7.1 Состав и структура выходных данных

Наименование данных	Обозначение	Структура данных	Способ вывода данных	Ограничения
Результат точности распознавания подписи нейронной сетью	predict	Число с плавающей точкой	Вызов метода predict в программе	От 0 до 1 включительно
Результат точности распознавания подписи нейронной сетью	accuracy	Число с плавающей точкой	Вызов метода accuracy в программе	От 0 до 1 включительно
Индекс владельца подписи в таблице базы данных	index	Целое число	Вызов метода classify_picture	Не меньше 0

### 7.2 Интерпретация выходных данных

Результатом работы программы является – **index** – индекс владельца подписи в таблице базы данных при её наличии (рисунок 7.2.1). Выходные данные выводятся на консоль компьютера.

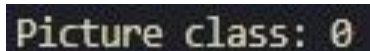


Рисунок 7.2.1 – Результат идентификации подписи