

Инструкция

по применению материалов для выполнения курсовых и самостоятельных работ

1. ОБЩИЕ СВЕДЕНИЯ

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://github.com/ElishaFlacon/signess> . Для его использования необходим компилятор Visual Studio Code или PyCharm, датасет со сканированными образами подписей и сводной таблицей данных, содержащей ФИО и индекс владельца подписи в лексикографическом порядке.

2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Назначение программного компонента

Задача программы - распознать рукописную подпись на сканированном изображении и идентифицировать ее владельца.

Для успешного решения поставленной задачи в общем случае выполняются следующие операции:

1. Создание датасета.
2. Определение области печати с подписью.
3. Выделение признаков подписи человека на документе, скрытой информационным шумом (печатью);
4. Подготовка изображений:
 - a. обрезка изображения по контуру;
 - b. изменение размеров изображения;
 - c. переворачивание изображения;
5. Удаление цифрового шума:
 - a. удаление текста;
 - b. удаление печати;

6. Классификация подписи.
7. Идентификация подписи по имеющейся базе данных с другими подписями людей.

2.2 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 приспособлять 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.

2.3 Функциональные условия применения

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.
- изображение не должно быть черно-белым.

2.4 Технические условия применения

Вычислительная машина должна использовать ОС Windows или 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.

3. ОБРАЩЕНИЕ К ПРОГРАММЕ

3.1 Точки входа в программу

Точками входа в программу являются начала модулей, запускаемых в ходе выполнения программы:

- инициация класса Dataset;
- инициация класса Autograph;
- инициация класса Document;
- инициация класса FedotCNN.

3.2 Базовые функции

На рисунке 4.2.2 представлен основной модуль системы, отвечающий за сборку датасета и вызов метода классификации изображения. Для этого подключаем библиотеки `signess` и `inscrib`, более полные названия приведены ниже на рисунке 4.2.1. 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`. Остается только собрать пайплайн обучения модели и вызвать метод классификации изображения.

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

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

Рисунок 4.2.1 – 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 4.2.2 – Конфигурация и вызов главной функции программы в основном модуле

3.3 Пайплайны на основе функций компонента.

На рисунке 4.3.1 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 4.3.1 – Пайплайн обучения модели

4. УСТАНОВКИ И ПРОВЕРКА ПРОГРАММЫ

4.1 Установка и настройка программы

Программа представляет собой библиотеку для языка программирования Python. Ее можно установить при помощи пакетного менеджера pip командой `pip install signess`.

Для тестового запуска необходимо скопировать содержимое директории из репозитория библиотеки (<https://github.com/ElishaFlacon/signess>), предварительно установив саму библиотеку и запустить файл `example.py`. В нем находится пример кода для запуска программы.

4.2 Модульные и интеграционные тесты

В директории предусмотрены тестовые выборки различных подписей, на которых можно проверить работу библиотеки.

После скачивания библиотеки и установки необходимых файлов директории можно будет провести тестирование.

5. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

5.1 Входные данные

В таблице 5.1.1 приведены состав и структура входных данных.

Таблица 5.1.1

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 документов 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	

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.

5.2 Выходные данные

В таблице 5.2.1 приведены состав и структура выходных данных.

Таблица 5.2.1

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

Результатом работы программы является – **index** – индекс владельца подписи в таблице базы данных при её наличии (рисунок 5.2.1). Выходные данные выводятся на консоль компьютера.

```
Picture class: 0
```

Рисунок 5.2.1 – Результат идентификации подписи

6. СООБЩЕНИЯ

При запуске программы для удобства пользователя будет выведен интерфейс (см рисунок 6.1

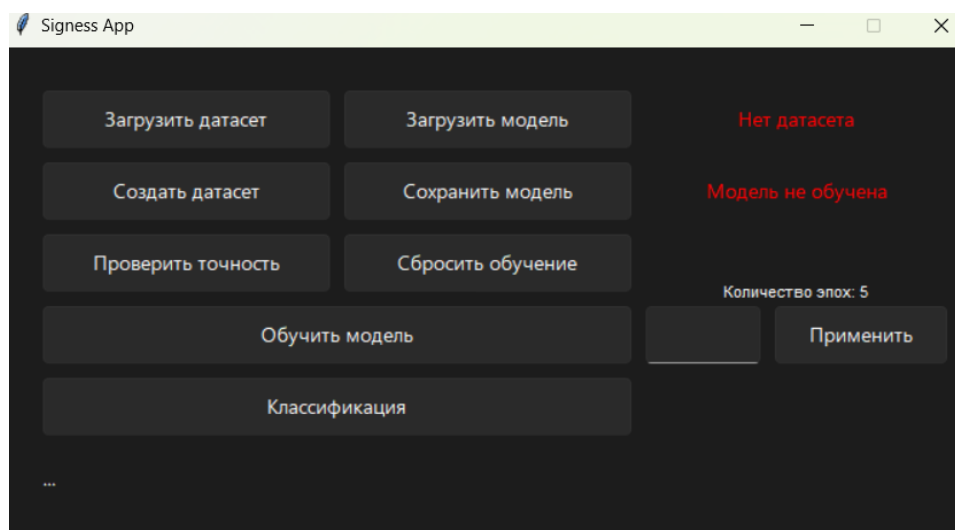


Рисунок 6.1 – Меню приложения

После выбора пользователем опции “Загрузить датасет”, а также выбора пути к датасету, ему выведется сообщение, показанное на рисунке 6.2.

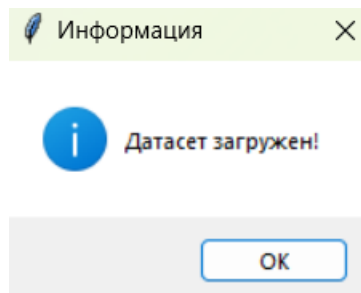


Рисунок 6.2 – Информация о загрузке датасета

Если на этапе выбора пути пользователь отменит операцию, то ему будет выведено сообщение, показанное на рисунке 6.3.

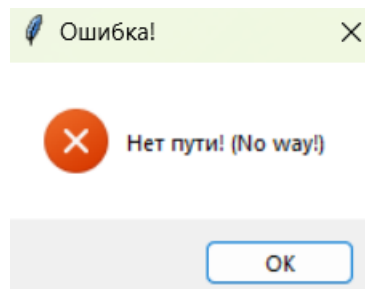


Рисунок 6.3 – Ошибка выбора пути

Если пользователь выберет опцию “Создать датасет”, и укажет путь к папке с изображениями, то программа оповестит его о начале работы (см. рисунок 6.4), а после о ее завершении (см. рисунок 6.5).

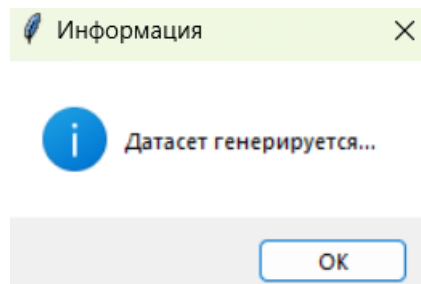


Рисунок 6.4 – Начало генерации датасета

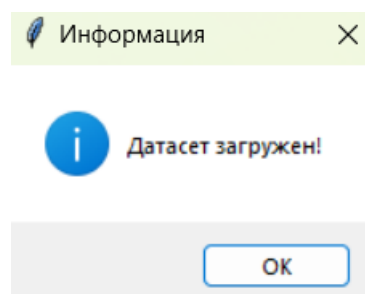


Рисунок 6.5 – Окончание генерации датасета

О том, что датасет загружен пользователь так же может узнать, посмотрев на главное меню (см. рисунок 6.6).

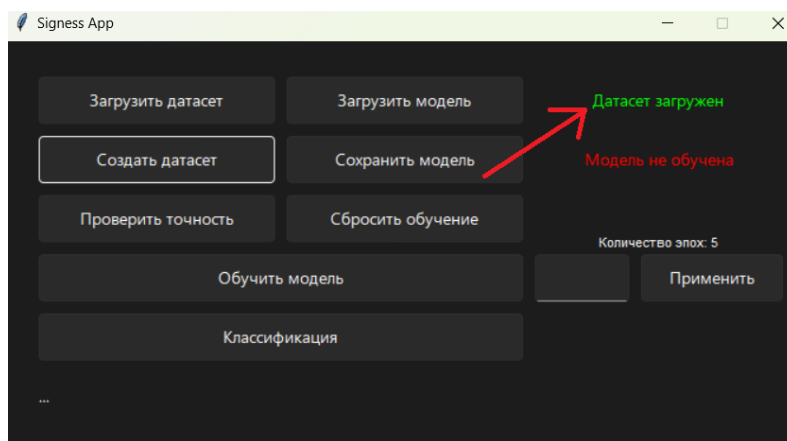


Рисунок 6.6 – Указатель готовности датасета

После выбора опции “Обучить модель”, приложение оповестит о том, что модель обучается (см. рисунок 6.7).

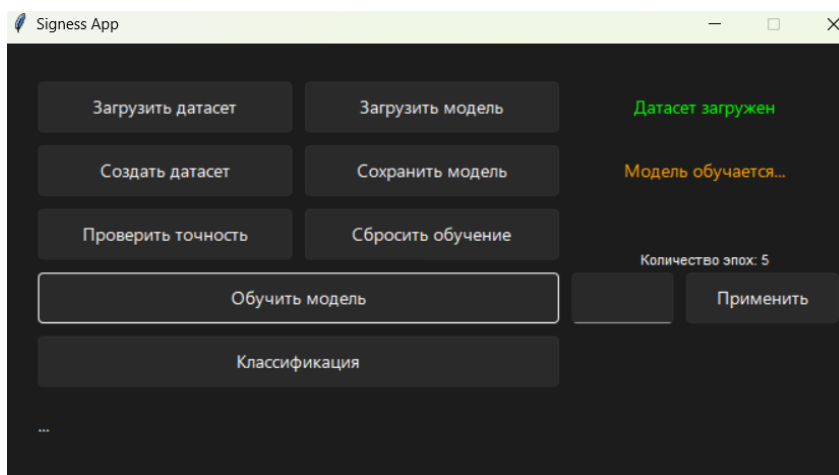


Рисунок 6.7 – Процесс обучения модели

И после обучения модели программа сообщит пользователю об этом (см. рисунок 6.8).

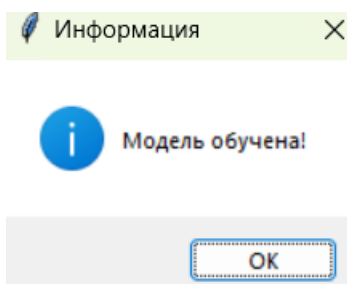


Рисунок 6.8 – Информация об обучении модели

Так же о том, что модель обучена можно узнать в главном меню (см. рисунок 6.9).

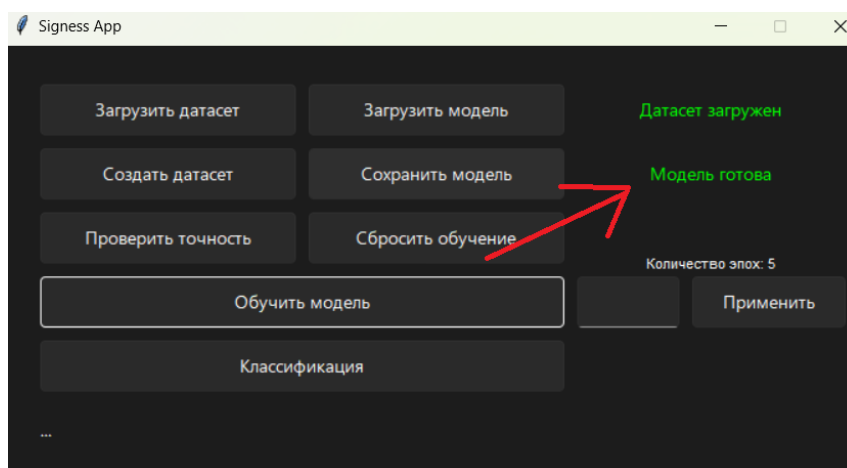


Рисунок 6.9 – Указатель готовности модели

При выборе опции “Проверить точность” необходимо будет указать путь к датасету, после чего программа проверит на нем точность обученной модели, и сообщит об итогах пользователю (см. рисунок 6.10).

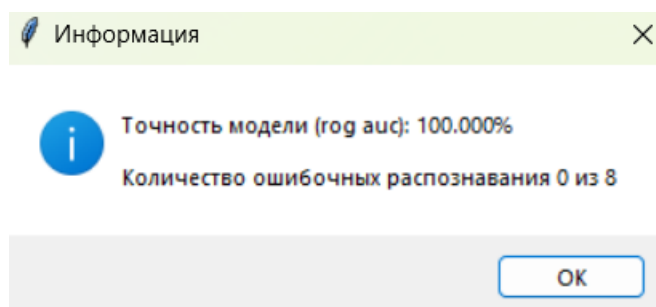


Рисунок 6.10 – Информация о точности

Опция “Сбросить обучение” удалит обучение модели. При завершении пользователь получит сообщение, показанное на рисунке 6.11.

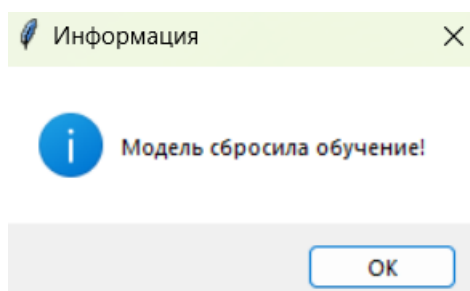


Рисунок 6.11 – Информация о сбросе обучения

Опция “Сохранить модель” создаст, в выбранной пользователем папке, папку с моделью, которая сейчас готова к использованию.

После чего подобные папки можно использовать в опции “Загрузить модель”, которая позволит использовать выбранную модель в приложении. Программа сообщит пользователю необходимые требования к модели, загружаемой таким образом (см. рисунок 6.12).

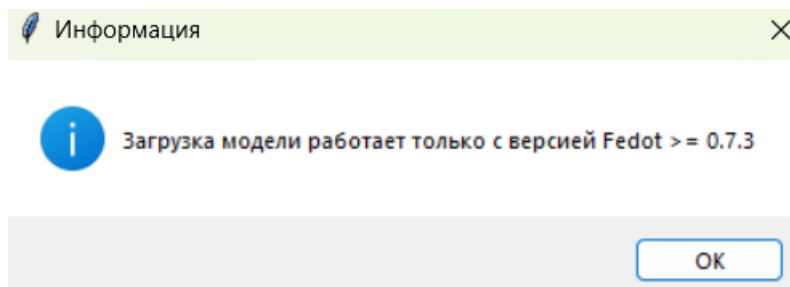


Рисунок 6.12 – Информация о необходимой версии FEDOT

Опция “Классификация” покажет принадлежность к классам, заданным в обученной модели, выбранного пользователем файла (см. рисунок 6.13).

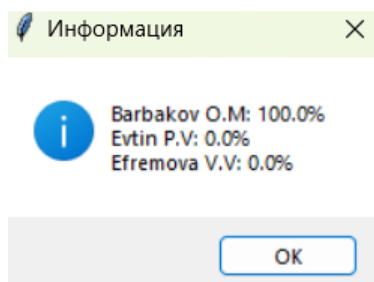


Рисунок 6.13 – Классификация файла

После этого в главном меню можно будет увидеть класс файла (см. рисунок 6.14).

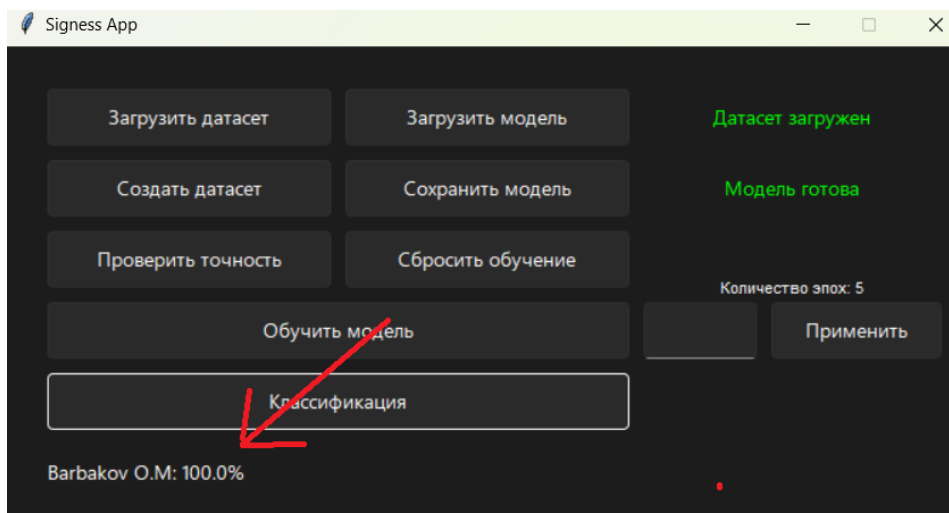


Рисунок 6.14 – Класс файла в главном меню

Если файл будет выбран некорректно, программа оповестит об этом пользователя. (см. рисунок 6.15).

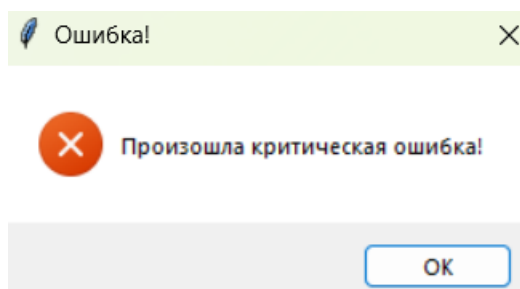


Рисунок 6.15 – Ошибка выбора файла

Перечень тем курсовых и самостоятельных работ

Приводится описание нескольких проблемных кейсов, решение которых возможно с применением библиотеки SRS (Signess). По каждому кейсу может быть сформулировано несколько конкретных работ.

Кейс 1. Архив города Т.

Описание проблемной области.

Городской архив, в котором хранятся все документы бюджетных учреждений.

Архив хранит в себе огромный объем документов, часть из которых оцифрованы. Бюджет архива ограничен и рассчитан только на его содержание.

В архив поступают запросы на предоставление всех документов, подписанных определенным лицом. Учитывая объем хранимых данных, сжатые сроки ответа на запрос, неполную оцифровку документов и отсутствие единой систематизации хранимых данных, сотрудники архива занимаются поиском необходимых документов вручную, что полностью останавливает работу над другими блоками задач.

Возможным результатом решения обозначенной проблемы является создание единого реестра всех документов. Реестр может содержать в себе не только информацию о содержании документа и его происхождение, но и дополнительные настраиваемые теги и информацию о подписавшем лице.

Преимущества данного метода: исключение возможности ошибки, оперативный доступ ко всей информации и простота поиска документов.

Недостатки: внедрение такого реестра требует колоссальных трат бюджета и прилагаемых усилий, а при текущем объеме хранимых документов, перенос всей информации в реестр становится невыполнимой задачей. В случае успешного внедрения данной системы внесение изменений в формат хранимых данных становится невозможным или слишком затратным.

Задание.

Учитывая ограниченность в бюджете и необходимость экстренного инструмента для решения существующей проблемы, возможно применение искусственной нейронной сети, способной быстро и качественно обработать большой объем данных и определить все документы подписанные конкретным лицом.

Signess - универсальная система, на основе алгоритмов искусственного интеллекта способная быстро и с высокой долей точности классифицировать подпись скрытую визуальным шумом, таким как текст или круглая печать. Одной из планируемых функций является пакетная обработка документов с последующей фильтрацией результатов классификации, позволяющая в

кратчайшие сроки определить, в каких документах находится подпись необходимого лица и сохранить полученный результат в файл для дальнейшей работы.

Рекомендуемый алгоритм внедрения.

Для внедрения Signess в работу архива города Т потребуется совсем немного усилий. При этом, систему можно настроить как на распознавание всех подписей, так и на распознавание какой-то конкретной.

Подготовительный этап состоит из следующих шагов:

1. Оцифровать все документы за необходимый период, при этом важна лишь оцифровка страницы документа с подписью.
2. Скачать и установить Signess на любой рабочий компьютер.
3. Собрать 10 документов содержащих подпись необходимого человека.
4. С помощью внутреннего интерфейса программы создать датасет на основе собранных документов.
5. Обучить модель нейронной сети на собранном датасете.

Программа полностью готова к использованию. Для этого необходимо установить флажок в поле «Фильтр» и указать всех необходимых людей, чьи подписи будут найдены, после чего нажать на кнопку «Классификация» и выбрать все документы, в которых будет производиться поиск.

Ожидаемые результаты.

Результатом классификации документов с последующей фильтрацией является перечень названий документов и фамилий лиц, чья подпись в них фигурирует, из тех что были указаны в фильтре, при этом каждому документу соответствует единственная фамилия человека, его подписавшего.

Внедрение Signess позволит оперативно обрабатывать поступающие запросы на предоставление всех документов, подписанных определенным лицом, при этом на выполнение данной задачи будет достаточно несколько часов работы одного сотрудника, что позволит сконцентрировать работу архива на выполнении более важных задач.

Кейс 2. Учет материальных ценностей.

Описание проблемной области.

Бухгалтерия крупного предприятия, в котором периодически происходит постановка на учет материальных ценностей в больших объемах и разным лицам.

Предприятие перед каждым крупным мероприятием оформляет закупку необходимых материалов и ценностей у множества различных поставщиков. Материально ответственным лицом за конкретные ценности назначается тот человек, чья подпись стоит в товарной накладной. Учитывая большой объем подписываемых товарных накладных, бухгалтерии становится сложно уследить за своевременной постановкой на учет материальных ценностей, за счет чего приходится многократно перечитывать документы и перепроверять кто их подписал.

Вариантом решения обозначенной проблемы является внедрение электронного документооборота с полным циклом закупки и учета в электронном виде.

Данный метод позволит качественно следить за всеми процессами, сопутствующими приобретению материальных ценностей и не только, при этом работая автономно и автоматически.

С другой стороны, для работы в полном электронном документообороте необходимо выполнение следующих условий: переход всех подрядчиков на электронный документооборот; перенос всех внутренних процессов компании в электронный формат; все материально ответственные лица должны быть зарегистрированы в системе электронного документооборота с правом использования электронной подписи. Учитывая все перечисленные факторы, внедрение такой системы крайне маловероятно избавит бухгалтерию от подписания товарных накладных на бумажном носителе.

Задание.

Для решения поставленной проблемы с минимальными трудозатратами возможно внедрение системы, способной пакетно проанализировать множество файлов, сопоставляя их с лицом, подписавшим соответствующий документ.

Signess - универсальная система, на основе алгоритмов искусственного интеллекта способная быстро и с высокой долей точности классифицировать подпись скрытую визуальным шумом, таким как текст или круглая печать. Одной из планируемых функций является пакетная обработка документов, позволяющая в кратчайшие сроки определить, кто подписал каждый из выбранных документов и сохранить полученный результат в файл для дальнейшей работы.

Рекомендуемый алгоритм внедрения.

Для внедрения Signess в процессы компании достаточно всего нескольких часов, при этом нет необходимости останавливать текущие внутренние процессы и менять их структуру.

Достаточно произвести ряд простых действий:

1. Собрать по 10 подписанных документов от каждого материально ответственного лица.
2. Скачать и установить Signess.
3. С помощью внутреннего интерфейса программы создать датасет на основе собранных документов.
4. Обучить модель нейронной сети на собранном датасете.

Программа полностью готова к использованию. При необходимости узнать, кем подписан документ (или несколько документов), достаточно нажать кнопку «Классификация» и выбрать необходимые файлы.

Ожидаемые результаты.

Результатом классификации документов является перечень названий этих документов и фамилий материально ответственных лиц, чья подпись в них фигурирует, при этом каждому документу соответствует единственная фамилия человека, его подписавшего.

Внедрение Signess позволит сэкономить трудозатраты бухгалтерии на многочисленную проверку документов. Так с минимальным использованием ресурсов будет решена одна из проблем, занимающая временной ресурс, что позволит перераспределить освободившееся время сотрудников на более важные задачи.

Кейс 3. Отдел обработки корреспонденции.

Описание проблемной области.

В крупном государственном высшем учебном заведении существует отдел, занимающийся обработкой входящей и исходящей корреспонденции.

Ежедневно в адрес этого отдела поступают сотни писем и документов. Задачей отдела является распределение входящей корреспонденции по исполнителям.

Укомплектованный отдел из 4 сотрудников занимается в большей мере отправкой писем из конкретной организации конкретному исполнителю. Учитывая большое количество входящей корреспонденции, это является их основной работой, что является нецелесообразным использованием человеческого ресурса, а также бюджета на содержание отдела.

На сегодняшний день проблема распределения писем по исполнителям может быть решена путем автоматической пересылки электронных писем от конкретного отправителя конкретному исполнителю.

Данный метод может полностью автоматизировать работу с входящими электронными письмами. Но в реальности письма из одной организации могут приходить с разных электронных адресов, а также все еще значительная часть корреспонденции поступает на бумажных носителях. Поэтому внедрение данной системы практически не уменьшит нагрузку на отдел обработки корреспонденции.

Задание.

Для уменьшения нагрузки на сотрудников возможно внедрение системы Signess, способной быстро и качественно обработать большой объем данных и определить все документы подписанные конкретным лицом, а учитывая, что круг лиц, подписывающих исходящую корреспонденцию, ограничен, то можно абсолютно точно определить, из какой организации пришло письмо. Одной из планируемых функций Signes является классификация документов не только по подписывающему лицу, но и группировка по компании, в которой работают несколько лиц, чьи подписи мы распознаем.

Рекомендуемый алгоритм внедрения.

Signess может очень легко внедриться в существующую систему, не прерывая текущую работу. Для этого понадобится:

1. Собрать по 10 подписанных документов от каждого представителя каждой компании.
2. Скачать и установить Signess.
3. С помощью внутреннего интерфейса программы создать датасет на основе собранных документов.
4. Обучить модель нейронной сети на собранном датасете.

Программа полностью готова к использованию. При необходимости узнать, кем подписан документ (или несколько документов), достаточно нажать кнопку «Классификация» и выбрать необходимые файлы.

Ожидаемые результаты.

Результатом классификации документов является перечень названий этих документов и фамилий представителей компаний с указанием компании, чья подпись в них фигурирует, при этом каждому документу соответствует единственная фамилия человека, его подписавшего.

Внедрение Signess позволит перераспределить трудовые ресурсы таким образом, чтобы сотрудники занимались более важными задачами, чем распределение писем. Это позволит сэкономить бюджет и повысить производительность труда.