

Аналитическая система почерковедческой идентификации авторов документов

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

Данный документ представляет собой руководство по установке Аналитической системы почерковедческой идентификации авторов документов (АСПИАД).

Документ «Инструкция по установке программного обеспечения» содержит сведения: о технических и программных средствах, настройке сервера, установке приложения, настройке базы данных, запуске приложения и создании службы для постоянной работы приложения.

СОДЕРЖАНИЕ

1. Требования к техническим и программным средствам	4
2. Инструкция по установке и запуску программы	9
2.1. Настройка сервера.....	9
2.2. Установка приложения	9
2.3. Конфигурационные файлы.....	10
2.4. Настройка базы данных.....	9
2.5. Запуск приложения	10
2.6. Создание службы для постоянной работы приложения.....	12

1. ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ И ПРОГРАММНЫМ СРЕДСТВАМ

Минимальный комплекс технических средств для проведения анализа рукописных материалов должен включать в себя:

- два сервера обработки работ (бланков), для проведения автоматического и ручного анализа;
- сервер базы данных, для хранения информации о ходе проведения анализа и информации о результатах анализа.

Требования к техническим средствам, необходимым для работы Программы, представлен в Таблице 1.

Таблица 1 – Описание комплекса технических средств

Оборудование	Количество	Требования	
Сервер обработки работ	2	Процессор	Intel Core i7 или аналогичное по производительности решение
		Графическая карта	8 × Видеокарта NVIDIA GeForce GTX1080 8GB GDDR5X или аналогичное по производительности решение
		Оперативная память	4 x 16 Гб DDR4
		Объем хранилища	1 Тб
		Сетевая карта	1 Gigabit Ethernet
Сервер базы данных	1	Процессор	Intel Xeon 4 Core или аналогичное по производительности решение
		Оперативная память	4 x 64 Гб DDR4
		Объем хранилища	1 Тб
		Сетевая карта	1 Gigabit Ethernet

Комплекс программных средств, совместимость с которыми обеспечивает Программа, включает в себя:

- Операционная система: Ubuntu (рекомендуется версия 18.04), Linux, Microsoft Windows;
- Системное ПО: Драйвер видеокарты NVIDIA;

- Система управления базами данных: MariaDB;
- Среда разработки: Visual Studio Code;
- Язык разработки: Python;
- Браузер: Mozilla Firefox, Google Chrome, Яндекс.Браузер.

Кроме того, для корректной работы с Программой рекомендуется установить следующие библиотеки Python:

- OpenCV;
- PyTorch;
- Flask – WTForms;
- Pyzbar;
- MarkupSafe;
- Pillow (PIL);
- Marshmallow;
- Pandas;
- Numpy;
- Scipy;
- Matplotlib;
- scikit-learn;
- keras;
- tensorflow.

Описание использованных библиотек Python представлено в Таблице 2.

Таблица 2 – Описание использованных библиотек

Наименование библиотеки	Лицензия	Комментарий
OpenCV	BSD	Библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD.
PyTorch	BSD	Библиотека машинного обучения для языка Python с открытым исходным кодом, созданная на базе Torch. Используется для решения различных задач: компьютерное зрение, обработка естественного языка.

Flask - WTForms	BSD	Фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Относится к категории микрофреймворков — минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.
pyzbar	MIT	Библиотека для обнаружения и декодирования штрих-кодов и QR-кодов различных типов. pyzbar распространяется под лицензией MIT.
MarkupSafe	BSD	MarkupSafe экранирует символы, поэтому текст безопасно использовать в HTML и XML. Символы, имеющие специальные значения, заменяются, чтобы они отображались как действительные символы. Это смягчает атаки с использованием инъекций, означает, что ненадежный пользовательский ввод может безопасно отображаться на странице.
Pillow (PIL)	Python Imaging Library license	Библиотека изображений Python, или PIL (Python Imaging Library) нужна для обработки растровой графики в Python.
marshmallow	MIT	Библиотека, независимая от ORM / ODM / Framework, для преобразования сложных типов данных, таких как объекты, в и из собственных типов данных Python. Библиотека для форматирования данных и проверки схем, используется для функций: <ul style="list-style-type: none"> • проверка входных данных; • десериализация входных данных в объекты уровня приложения; • сериализация объектов уровня приложения для примитивных типов Python.
pandas	BSD	Программная библиотека на языке Python для обработки и анализа данных. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Основная область применения — обеспечение работы в рамках среды Python

		не только для сбора и очистки данных, но для задач анализа и моделирования данных, без переключения на более специфичные для статобработки языки.
numpy	BSD	<p>Библиотека с открытым исходным кодом для языка программирования Python. Возможности:</p> <ul style="list-style-type: none"> • поддержка многомерных массивов (включая матрицы); • поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами. <p>Библиотека NumPy предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами.</p>
scipy	BSD	<p>Библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов.</p> <p>Возможности:</p> <ul style="list-style-type: none"> • поиск минимумов и максимумов функций; • вычисление интегралов функций; • поддержка специальных функций; • обработка сигналов; • обработка изображений; • работа с генетическими алгоритмами; • решение обыкновенных дифференциальных уравнений.
matplotlib	Matplotlib licence	<p>Библиотека на языке программирования Python для визуализации данных двумерной (2D) графикой (3D графика также поддерживается).</p> <p>Пакет поддерживает многие виды графиков и диаграмм:</p> <ul style="list-style-type: none"> • графики (line plot); • диаграммы разброса (scatter plot); • столбчатые диаграммы (bar chart) и гистограммы (histogram); • круговые диаграммы (pie chart);

		<ul style="list-style-type: none"> • ствол-лист диаграммы (stem plot); • контурные графики (contour plot); • поля градиентов (quiver); • спектральные диаграммы (spectrogram).
scikit-learn	Новая лицензия BSD	Является свободным программным обеспечением машинного обучения библиотека для Python языка программирования. Она предоставляет широкий выбор алгоритмов обучения с учителем и без учителя. Обучение с учителем предполагает наличие размеченного датасета, в котором известно значение целевого признака. В то время как обучение без учителя не предполагает наличия разметки в датасете — требуется научиться извлекать полезную информацию из произвольных данных. Одно из основных преимуществ библиотеки состоит в том, что она работает на основе нескольких распространенных математических библиотек, и легко интегрирует их друг с другом.
keras	MIT	Открытая нейросетевая библиотека, написанная на языке Python. Нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой. Эта библиотека содержит многочисленные реализации широко применяемых строительных блоков нейронных сетей, таких как слои, целевые и передаточные функции, оптимизаторы, и множество инструментов для упрощения работы с изображениями и текстом.
tensorflow	Apache License 2.0	Открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия.

2. ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ЗАПУСКУ ПРОГРАММЫ

В данном разделе представлена инструкция по установке и запуску программы на операционной системе Ubuntu 18.04 LTS.

2.1. Развертывание базы данных

Для развертывания базы данных тестового приложения требуется установить СУБД MariaDB.

СУБД MariaDB является свободно-распространяемым (бесплатным) программным обеспечением. В Ubuntu 18.04 MariaDB включена в репозитории пакетов АРТ по умолчанию.

Чтобы установить его, обновите индекс пакета на вашем сервере с помощью apt:

```
sudo apt update
```

Затем установите пакет:

```
sudo apt install mariadb-server
```

Убедитесь, что MariaDB запущена с помощью systemctl start команды:

```
sudo systemctl start mariadb.service
```

2.2. Настройка сервера

Для установки драйверов видеокарты требуется ввести в терминале Ubuntu команды:

```
sudo apt install ubuntu-drivers-common
sudo add-apt-repository ppa:graphics-drivers/ppa
sudo apt purge nvidia*
ubuntu-drivers devices
```

Далее выбрать из полученного списка версию драйвера, в строке с которой содержится «recommended» и произвести установку командой*:

```
sudo apt install nvidia-driver-415
```

* номер версии в приведенном примере (415) может отличаться от рекомендованного

После чего требуется установить CUDA и CUDNN командами:

```
sudo apt install cuda-10-0
sudo apt install libcudnn7
```

2.3. Установка приложения

Для установки приложения требуется иметь на компьютере среду разработки Python 3.6. Среда разработки Python является свободно-распространяемым (бесплатным) программным обеспечением, которая включена в Ubuntu 18.04.

Для того, чтобы проверить установленную версию Python, откройте окно терминала и введите следующее:

```
python3 --version
```

Если у вас установлен Python 3.6, то в ответе будет содержаться номер версии.

Если Python 3.6. не установлен, можно установить Python 3.6 из Personal Package Archive J Fernyhough (PPA). Для этого требуется установить следующие требования:

```
apt-get install software-properties-common python-software-properties
```

Далее выполнить следующую команду, чтобы добавить PPA:

```
# add-apt-repository ppa:jonathonf/python-3.6
Press [ENTER] to continue or ctrl-c to cancel adding it
```

Требуется нажать клавишу «ВВОД», чтобы продолжить и обновить репозитории:

```
apt-get update
```

После этого установить Python версии 3.6:

```
apt-get install python3.6
```

После установки, вы можете проверить установленную версию командой:

```
python3 --version
```

После установки или проверки наличия установленного Python 3.6 необходимо распаковать содержимое архива «ASPIAD.zip» на сервере (var/ASPIAD) и установить пакетный менеджер pip и npm командами:

```
apt install python3-pip
python3 -m pip install --upgrade pip
python3 -m pip install -U setuptools
apt install npm
```

Также требуется установить зависимости приложения на сервере с помощью менеджера пакетов pip командами:

```
pip3 install -r requirements.txt
apt-get install zbar-tools
```

2.4. Конфигурационные файлы

В файле конфигурации «paths.py» необходимо указать корректные реквизиты подключения к базе данных для каждого из режимов работы приложения, развертывание которой выполнено в рамках п. 1.1.:

Описание файла конфигурации «paths.py» (var/ASPIAD/paths.py):

Переменная	Назначение
------------	------------

DB_TEST_CONN	Строка для подключения к БД для тестов
DB_DEV_CONN	Строка для подключения к БД для разработки
DB_PROD_CONN	Строка для подключения к удаленной БД

Формат строки для подключения:

```
mariadb+mariadbconnector://{имя пользователя}:{пароль}@{ip хоста}/{база данных}?charset=utf8mb4
```

```
from pathlib import Path

ROOT_PATH = Path(__file__).absolute().parent
app_path = ROOT_PATH / "app"
static_path = app_path / "static"
tables_path = static_path / "tables"
data_path = ROOT_PATH / "data"
datasets_path = ROOT_PATH / "datasets"
images_path = data_path / "images"
models_path = data_path / "models"
tests_path = ROOT_PATH / "tests"
logs_path = ROOT_PATH / "logs"

paths = [app_path, static_path, tables_path, data_path, datasets_path, images_path,
         models_path, tests_path, logs_path]
for path in paths:
    if not path.exists():
        path.mkdir(parents=True)

DB_TEST_CONN = 'mariadb+mariadbconnector://test:testing@10.32.200.172:3306/test?charset=utf8mb4'
DB_DEV_CONN = 'mariadb+mariadbconnector://dev:development@10.32.200.172:3306/dev?charset=utf8mb4'
DB_PROD_CONN = 'mariadb+mariadbconnector://prod:production@10.32.200.172:3306/web?charset=utf8mb4'
```

Рисунок 1 — Ввод реквизитов подключения к базе данных

Далее в файле конфигурации «config.py» необходимо указать настройки директорий и поддиректорий для хранения изображений, отчетов и почтовые настройки для каждого из режимов работы приложения, развертывание которой выполнено в рамках п. 1.1.

Описание файла конфигурации «config.py» (var/ASPIAD/app/config.py):

Переменная	Назначение
Config.MAIL_*	Настройки почты для рассылки кодов для сброса пароля
Config.STORAGE_DIR	Директория, в которой хранятся обработанные бланки
Config.REPORTS_DIR	Директория, в которой хранятся отчеты
*Config.BLANKS_SRC	Поддиректория Config.STORAGE_DIR с объединенными рукописными документами авторов
*Config.BLANKS_DST	Поддиректория Config.STORAGE_DIR с объединенными рукописными документами авторов, содержащими выделенные области, определённые как несанкционированные изменения

2.5. Запуск приложения

После корректировки файлов конфигурации необходимо запустить тестовое приложение на компьютере.

В каждой новой shell-сессии на сервере с системой нужно задать значения переменных окружения (если будет использоваться команда flask):

```
export FLASK_APP=web.py
export FLASK_DEBUG=0
export FLASK_CONFIG=production
export FLASK_ENV=production
export PYTHONPATH=$ASPIAD:$ASPIAD/libs/ssd_keras flask shell
```

FLASK_APP задает путь к файлу var/ASPIAD/web.py.

FLASK_DEBUG включает дебаг режим. В этом режиме изменения в коде автоматически подгружаются в web-приложение и активируется дебаггер.

FLASK_CONFIG задает, какая строка для подключения будет использована.

FLASK_CONFIG	Строка для подключения
testing	DB_TEST_CONN
development	DB_DEV_CONN
production	DB_PROD_CONN

FLASK_ENV задает использование модели для распознавания почерка, как подозрительного.

FLASK_ENV	Использование модели
production	+
development	-

2.6. Создание службы для постоянной работы приложения

Для создания службы для постоянной работы приложения необходимо:

Создать файл с помощью текстового редактора (например, nano):

```
nano /etc/systemd/system/aspiad.service
```

Далее указать в созданном файле параметры службы (при необходимости изменить путь по папки с приложением):

```
[Unit]
Description=ASPIAD
After=network.target
```

```
[Service]
User=root
WorkingDirectory=/home/developer/ASPIAD
Environment="FLASK_APP=/home/developer/ASPIAD/web.py"
Environment="FLASK_DEBUG=0"
Environment="FLASK_CONFIG=production"
Environment="FLASK_ENV=production"
Environment="PYTHONPATH=/home/developer/ASPIAD/:/home/developer/ASPIAD/libs/ssd_keras flask shell"
ExecStart= /bin/bash -c 'flask run --host=0.0.0.0'
Restart=always

[Install]
WantedBy=multi-user.target
```

После чего сохранить файл и запустить службу с помощью следующих команд:

```
systemctl daemon-reload
systemctl start aspiad.service
systemctl enable aspiad.service
```

Можно проверить статус службы командой:

```
systemctl status aspiad.service
```

В случае успешной работы службы отобразится сообщение с текущим статусом службы (active):

```
● aspiad.service - ASPIAD
   Loaded: loaded (/etc/systemd/system/aspiad.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-11-30 10:52:54 UTC; 23h ago
     Main PID: 12416 (flask)
        Tasks: 5 (limit: 4915)
      CGroup: /system.slice/aspiad.service
              └─12416 /usr/bin/python3 /usr/local/bin/flask run --host=0.0.0.0
```